Wonderware® FactorySuite™

InTouch™ 用户指南

修订版 **A**

最后修订日期：**2002** 年 **11** 月 **20** 日

Invensys Systems, Inc.

Invensys Systems, Inc.
33 Commercial Street
Foxboro, MA 02035
(949) 727-3200
http://www.wonderware.com

目录

第 3 章： 使用 InTouch 安全性........................169

第 4 章： 使用 InTouchView 195

第 5 章： 建立分布式应用程序 201

第 6 章： 标记名字典 227

第 8 章： 在 InTouch 中创建 QuickScript 403

第 10 章： 报警 / 事件客户端...........................533

第 11 章： 报警实用程序 649

第 13 章： I/O 通讯 733

第 14 章： Terminal Services for InTouch 749

欢迎使用 InTouch

欢迎使用 Wonderware® InTouch®，对于在 Microsoft® Windows™ 2000 与 Windows NT 操作系统上创建人机界面（Human-Machine Interface，简称 HMI）应用程序而言，这是一套简单快捷的应用程序。InTouch 是 Wonderware FactorySuite™ 的一个组成部分。InTouch 广泛应用于包括食品加工、半导体、石油、汽车、化工、制药、纸浆和造纸、交通等在内的全球众多纵向市场。

如需有关所支持的操作系统版本的详细信息，请参阅 “系统要求”。

通过使用 InTouch，您可以创建性能强劲、功能齐全的应用程序，充分利用 Microsoft Windows 的各项主要功能，包括 ActiveX 控件、 OLE、图形及联网等。通过添加自定义的 ActiveX 控件、向导、一般对象以及创建 InTouch QuickScript 扩展等，还可以充分扩展 InTouch 的功能。

InTouch 包含三个主要程序，它们分别是 “InTouch 应用程序管理器”、 WindowMaker™ 以及 WindowViewer™。

“InTouch 应用程序管理器”用于组织管理创建的应用程序。它也可以用于将 WindowViewer 配置成 NT 服务、为基于客户机和基于服务器的架构配置 “网络应用程序开发” (NAD)，以及配置 “动态分辨率转换” (DRC) 和 （或）分布式报警。DBDump™ 与 DBLoad™ 数据库实用程序也是从 “应用程序管理器”启动的。

WindowMaker 是一种开发环境，在其中可以使用面向对象的图形来创建富于动感的触控式显示窗口。这些显示窗口可以连接到工业 I/O 系统以及其它的 Microsoft Windows 应用程序。

WindowViewer 则是一种运行时环境，用于显示在 WindowMaker 中创建的图形窗口。WindowViewer 可以执行 InTouch QuickScript、执行历史数据记录与报告、处理报警记录与报告，并可以充当 DDE 与 SuiteLink? 通讯协议的客户端和服务器。

目录

- 第一次运行 InTouch
- 使用 InTouch 应用程序管理器

系统要求

要运行 InTouch，我们建议配备下列硬件与软件：

- 任何 IBM® 兼容 PC，配备 Pentium II 或更高级的处理器 （最低：单节点系统上为 400MHz，建议：1.2GHz 或更高主频）。
- 至少 2GB 空闲硬盘空间。
- 至少 256MB 随机访问内存 (RAM)，建议配备 512MB 的 RAM。

备注 我们建议每 5K 标记名使用 5MB 的 RAM。例如，支持 32K 标记名要求配备 32MB 的 RAM ；支持 60K 标记名要求配备 128MB 的 RAM。

- SVGA 显示适配器 （建议配备 2MB 的 RAM）。

备注 建议使用 800x600 或更高分辨率来查看 WindowMaker，以确保所有对话框均清晰可见。

- 定位设备。例如鼠标、轨迹球及触摸屏。
- Microsoft? Windows? 2000 Professional with Service Pack 1 或 Windows XP? Professional with Service Pack 3 操作系统。
- 对于使用 Windows 2000 操作系统，要实现 InTouch 的分布式功能，则必须安装与运行 Wonderware NetDDE。

备注 Wonderware FactorySuite InTouch 8.0 版 （或更高版本）不支持 Microsoft Windows 3.x 或 Microsoft Windows for Workgroups 或 Windows 9x 操作系统。

安装 InTouch

InTouch 使用 Wonderware FactorySuite 安装程序进行安装。InTouch 运行在 Microsoft Windows 2000 或 Windows XP 操作系统上。安装程序会创建所需的目录，然后将文件从光盘复制到硬盘。

备注 如需完整的 InTouch 安装说明，请参阅 InTouch “发行说明” (ITRELNOTES.TXT)。如需完整的 FactorySuite 安装说明，请参阅 FactorySuite “发行说明” (FSRELNOTES.TXT)。此外，您的联机 *FactorySuite 系统管理员指南*中也非常详尽地提供了 FactorySuite 软件包中大部分产品的安装说明。

从以前版本的 InTouch 升级

所有版本的 InTouch 均可从以前的版本轻松升级，您只要在两个对话框中分别确认是否要升级，以及是否已备份应用程序即可。您试图在 WindowMaker 或 WindowViewer 中打开先前版本的 InTouch 创建的应用程序时，便会出现这两个对话框。

备份旧应用程序

试图打开旧的应用程序时， WindowMaker 会检测出它是旧版本的，并显示一个**备份配置**对话框，提醒您在转换之前先行备份：



要更改缺省备份路径 （< 应用程序目录 >\Bak），请关闭**使用缺省备份路径**选项，然后在**备份路径**框中，输入用于保存应用程序备份的现有目录的路径。如果该目录不存在，则您必须先创建它，然后才能继续备份。

在**忽略文件**框中，您可以指定备份期间要忽略的任何文件。（缺省情况下，应用程序目录中的所有文件均会备份）。输入每个文件名，文件名之间用分号 (;) 隔开。

提示 您可以在文件名中使用标准通配符 （'*' 与 '?'）。

关于本手册

本手册由逻辑上相互连贯的多个章节组成，介绍建立 InTouch 应用程序所涉及的方方面面。它采用 “过程化”的格式编写而成，以带有序号的步骤阐述如何执行大多数功能或任务。

如果您联机阅读本手册，看到绿色文本时，可单击该文本以 “跳转”到所引用的部分或章节。跳转到另一节或另一章后，如果您想要返回原来的章节，请使用所提供的 “后退”选项。

提示 这些“提示”告诉您如何更简便快捷地完成某项功能或任务。

要熟悉 WindowMaker 开发环境及其工具，请阅读第 1 章“WindowMaker 程序元素”。要了解如何使用窗口、图形对象、向导及 ActiveX 控件等，请阅读第 2 章 “使用 WindowMaker”。如需有关运行时环境 (WindowViewer) 的详细信息，请阅读第 2 章 “使用 WindowMaker”。

此外，*InTouch 参考指南*还会为您提供具有一定深度的参考资料，向您介绍有关 InTouch QuickScript 语言与函数、系统标记名以及标记名**点域**的详细信息。

如需有关附加程序 SPC Pro 的详细信息，请参阅 *SPC Pro 用户指南*。

若需有关附加程序 Recipe Manager 的详细信息，请参阅 *Recipe Manager 用户指南*。

若需有关附加程序 SQL Access Manager 的详细信息，请参阅 *SQL Access Manager 用户指南*。

*FactorySuite 系统管理员指南*也为您提供了有关 FactorySuite 中的一般组件、系统要求、联网注意事项、产品集成、技术支持等方面的完整信息。

FactorySuite 软件包中还包含了全部 FactorySuite 组件的联机手册。

备注 要查看或打印联机手册，您必须安装 Adobe Acrobat Reader （4.0 或更高版本）。

假设

本手册假定您：

- 熟悉 Windows 2000 和 （或） Windows NT 操作系统工作环境。
- 了解如何使用鼠标、 Windows 菜单、选择选项及访问联机 “帮助”。
- 具备使用编程语言或宏语言的经验。要取得最佳效果，最好应理解一些编程概念，如变量、语句、函数及方法。

推荐参考读物

如需有关建立有效的人机界面的详细信息，我们建议您阅读：

The Windows Interface: An Application Design Guide, Microsoft Press, 1992.

Dreyfuss, Henry. *Symbol Sourcebook: An Authoritative Guide to International Graphic Symbols.* Van Nostrand Reinhold, 1984.

Laurel, Brenda. *The Art of Human-Computer Interface Design*. Addison-Wesley, 1990.

Norman, Donald A. *The Design of Everyday Things*. Doubleday, 1990.

Tufte, Edward. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

Chappell, David. *Understanding Active X and OLE - A Guide for Developer's and Managers*. Microsoft Press, Strategic Technology Series 1996.

技术支持

“Wonderware 技术支持”提供许多支持选项，可回答有关 Wonderware 产品及其具体实施的任何疑问。

在与技术支持部门联系之前，请参阅 *InTouch 用户指南*中相关的章节，以寻求您在使用 InTouch 系统时所遇问题的可能解决方案。如果您认为有必要求助于技术支持，请提供以下信息：

1. 软件序列号。
2. 运行的 InTouch 版本。
3. 所用操作系统的类型与版本。例如， Microsoft Windows NT 4.0 版 SP5 （或更高版本）工作站。
4. 所遇系统错误消息的准确描述。
5. Logger、 Microsoft Diagnostic utility (MSD) 或任何其它诊断应用程序提供的所有相关输出列表。
6. 尝试解决该问题的方法细节及其结果。
7. 有关如何重现该问题的细节。
8. 如果所遇问题是老问题，且该问题仍然存在，请提供指定的“Wonderware 技术支持”案例号。

如需有关 “技术支持”的详细信息，请参阅联机 *FactorySuite 系统管理员指南*。

FactorySuite 许可证

要查看 FactorySuite 系统许可证信息，请从 WindowMaker “帮助”菜单的**关于**对话框启动许可证查看程序。

要打开许可证实用程序

1. 在 WindowMaker 的**帮助**菜单上，单击**关于**。
2. 单击**查看许可证**。此时会出现**许可证实用程序 - LicView** 对话框。

如需有关许可证查看实用程序的详细信息，请参阅 *FactorySuite 系统管理员指南*。

第一次运行 InTouch

您第一次运行 INTOUCH.EXE 时，系统会自动创建一个 INTOUCH.INI 文件。此文件包含应用程序的系统缺省配置。在配置应用程序时，设置会写入 INTOUCH.INI 文件中。

自定义应用程序之后，在创建新的应用程序时，您可以将自定义的 INTOUCH.INI 文件复制到新应用程序的目录中。这样每次创建新的应用程序时，您就不必重新设置自定义参数了。

如需有关自定义应用程序的详细信息，请参阅第 2 章 “使用 WindowMaker”。

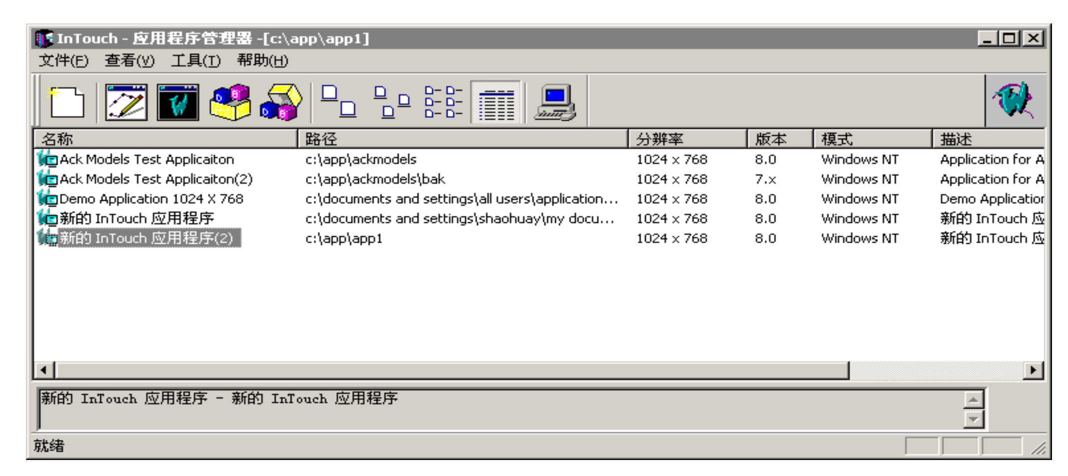
要第一次运行 InTouch

1. 启动 InTouch 程序 (intouch.exe)。此时会出现**欢迎使用 InTouch 应用程序管理器**对话框。
2. 单击**下一步**。此时会出现第二个**欢迎使用 InTouch 应用程序管理器**对话框，显示起始目录的缺省路径。例如， **C:\Documents and Settings\CPUName\My Documents\My InTouch Applications**。
3. 要指定一个不同的目录，请在输入框中输入目录路径，或单击**浏览**来查找目录。用户指定一个不同的目录时， InTouch 则缺省使用最近的用户访问的目录位置。
4. 单击**完成**。
5. 此时会出现 **InTouch - 应用程序管理器**，并自动搜索计算机中当前所有的 InTouch 应用程序。如果找到应用程序，则会在对话框中显示该应用程序的名称与图标。

要创建新的应用程序

1. 在**文件**菜单上，单击**新建**，或单击工具栏上的**新建**工具。此时会出现**创建新应用程序**向导。
2. 单击**下一步**。此时会出现第二个**创建新应用程序**向导。缺省情况下，系统将显示 InTouch 目录的路径，后跟 "**NewApp**"。
3. 在输入框中，输入要从中创建应用程序的目录路径，或单击**浏览**来查找目录。
4. 单击**下一步**。如果您指定的目录不存在，则会出现一个消息对话框，询问您是否要创建它。单击**确定**。此时会出现第三个**创建新应用程序**向导对话框。
5. 在**名称**框中，为新应用程序图标输入一个唯一的名称，应用程序列在 **InTouch 应用程序管理器**窗口中时，会显示该图标。
6. 在**描述**框中，给应用程序输入描述。（描述是可选项。不过，如果您输入了描述，则最多可以使用 255 个字符）。

7. 单击**完成**。此时会再次出现 **InTouch - 应用程序管理器**，显示图标与您为新应用程序指定的名称。



8. 要打开某个应用程序，请选择该应用程序并单击鼠标右键，然后在**文件**菜单中单击要给该应用程序使用的程序名；或者从列表中选择该应用程序，然后在工具栏中单击 WindowMaker 工具（WindowViewer 不能为新应用程序执行）。

提示 要快速打开应用程序，请双击其图标，或者选择它然后按 **Enter** 键。

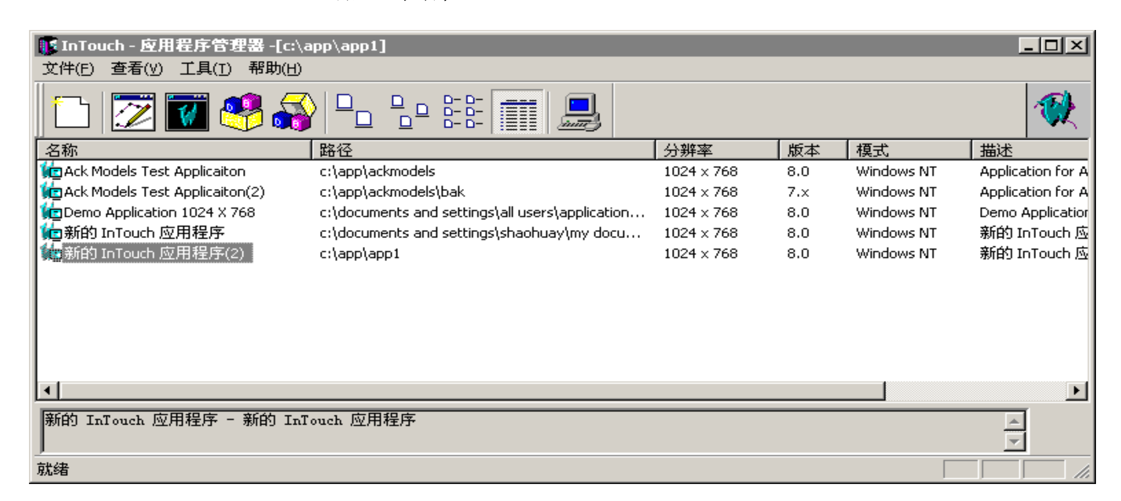
使用 InTouch 应用程序管理器

您可以使用 “InTouch 应用程序管理器”创建新应用程序、在 WindowMaker 或 WindowViewer 中打开现有的应用程序、删除应用程序，以及运行 InTouch DBDump 与 “DBLoad 标记名字典”实用程序。

如需有关 DBDump 与 DBLoad 程序的详细信息，请参阅第 6 章 “标记名字典”。

要运行 InTouch 应用程序管理器

启动 InTouch 程序 (intouch.exe)。此时会出现 “InTouch 应用程序管理器”对话框。



从列表中选择某个应用程序时，它的名称与描述文字将出现在屏幕底部的框中。如果您右击描述框，则会出现一个菜单，显示您可以应用于所选文本的各条命令。

在选择某个应用程序时并单击鼠标右键时，会出现一个菜单，从这个菜单中，您也可以执行 “InTouch 应用程序管理器”的多项菜单命令。例如：



要给应用程序的图标重命名，请在列表中右击该应用程序，然后单击**重命名**。键入新名称，然后按 **Enter** 键。

要删除某个应用程序的图标，请在列表中右击该应用程序，然后单击**删除**。此时会出现一个消息框，要求您确认删除。单击**是**从窗口中删除该应用程序，或单击**否**以取消删除。

备注 如果从列表中删除某个应用程序，它不删除您的文件或应用程序目录。

要查找应用程序

1. 在 “工具”菜单上，单击 “查找应用程序”。此时会出现 “搜索起始目录”对话框。

提示 要快速查找某个应用程序，请在窗口的空白区单击鼠标右键，然后在弹出菜单中单击**查找应用程序**。



2. 找到您要在其中搜索的应用程序目录，然后单击**确定**。

   此时会再次出现 "InTouch 应用程序管理器"，显示选定目录中找到的所有应用程序的图标。

要查看应用程序的属性

1. 从列表中选择应用程序。

2. 在**文件**菜单上，单击**属性**。此时会出现**属性**对话框。



要查看节点的属性

在工具菜单上，单击**节点属性**。此时会出现**节点属性**对话框。

提示 要快速访问**节点属性**对话框，请右击窗口的空白区，然后单击**节点属性**。



您可以使用此对话框来配置下列项目：

- WindowViewer 作为 NT 服务
- 网络应用程序开发 (NAD)
- 动态分辨率转换 (DRC)

如需有关详细信息，请参阅第 5 章 “建立分布式应用程序”。

要配置应用程序管理器的显示窗口

1. 在**查看**菜单上，单击要应用的命令，或右击任意列标题，或单击“应用程序管理器”窗口的空白区，或在所出现的弹出菜单上单击某个详细资料 （不是应用程序名）。



2. 指向**查看**，此时会出现下面的子菜单



3. 选择要应用的命令。

如需有关显示命令的详细信息，请参阅 “应用程序管理器的工具”。

应用程序管理器的工具

缺省情况下，初次运行 InTouch 时，会显示 “应用程序管理器”的工具栏与状态栏。

要隐藏工具栏

在**查看**菜单上，选择**工具栏**。要再次显示它，请重复此步骤。

要隐藏状态栏

在**查看**菜单上，选择**状态栏**。要再次显示它，请重复此步骤。

下面简要介绍 “应用程序管理器”工具栏上的每个工具：

工具	描述
	执行**文件**菜单上的**新建**命令，以创建一个新的应用程序。
	执行**文件**菜单上的 **WindowMaker** 命令，以打开 WindowMaker 中选定的应用程序。 **提示**要在 WindowMaker 中快速打开应用程序，请在显示列表中双击其图标，或选择它然后按 ENTER 键。
	执行**文件**菜单上的 **WindowViewer** 命令，以打开 WindowViewer 中选定的应用程序。
	执行**文件**菜单上的 **DBLoad** 命令，以运行用于加载 “标记名字典”输入文件的 DBLoad 实用程序。
	执行**文件**菜单上的 **DBDump** 命令，以运行用于提取应用程序的 “标记名字典”的 DBDump 实用程序。 如需有关 DBDump 与 DBLoad 程序的详细信息，请参阅第 6 章 “标记名字典”。

工具	描述
	执行**查看**菜单上的**大图标**命令，以便给列出的应用程序显示大图标。
	执行**查看**菜单上的**小图标**命令，以便给列出的应用程序显示小图标。
	执行**查看**菜单上的**列表**命令，以便将对话框更改为列表视图模式。
	执行**查看**菜单上的**详细**命令，以便将对话框更改为详细视图模式。
	执行**工具**菜单上的**节点属性**命令，以打开用于设置以下计算机属性的**节点属性**对话框： WindowViewer 作为 NT 服务 网络应用程序开发 (NAD) 动态分辨率转换 (DRC) 如需有关详细信息，请参阅第 5 章 “建立分布式应用程序”。

第 1 章

WindowMaker 程序元素

WindowMaker 是 InTouch 的开发环境。WindowMaker 图形用户界面符合 Windows 2000 与 Windows NT GUI 标准。WindowMaker 支持浮动和固定工具栏；在整个程序中到处都有提供鼠标右键菜单，供快速访问常用命令；提供支持 1.67 千万种颜色的可自定义调色板。（颜色支持仅受显卡性能的限制）。

WindowMaker 的 “应用程序浏览器”提供了一种功能强劲的图形化方法，供您浏览和配置 InTouch 应用程序。通过使用它，您可以轻易访问 WindowMaker 最常用的命令和功能，如所有的 Windows 命令、所有的配置命令以及所有的 InTouch QuickScript 编辑器。此外，“应用程序浏览器”还会显示所有安装的附加程序，如 SQL Access Manager、 SPC Pro 和 Recipe Manager，并为您提供了一种可自定义的应用程序启动工具。

您可以配置 “应用程序浏览器”来启动任何其它 FactorySuite 程序或 Windows 程序，以便在 HMI 配置、“I/O 服务器”配置及控件配置之间快速切换。

目录

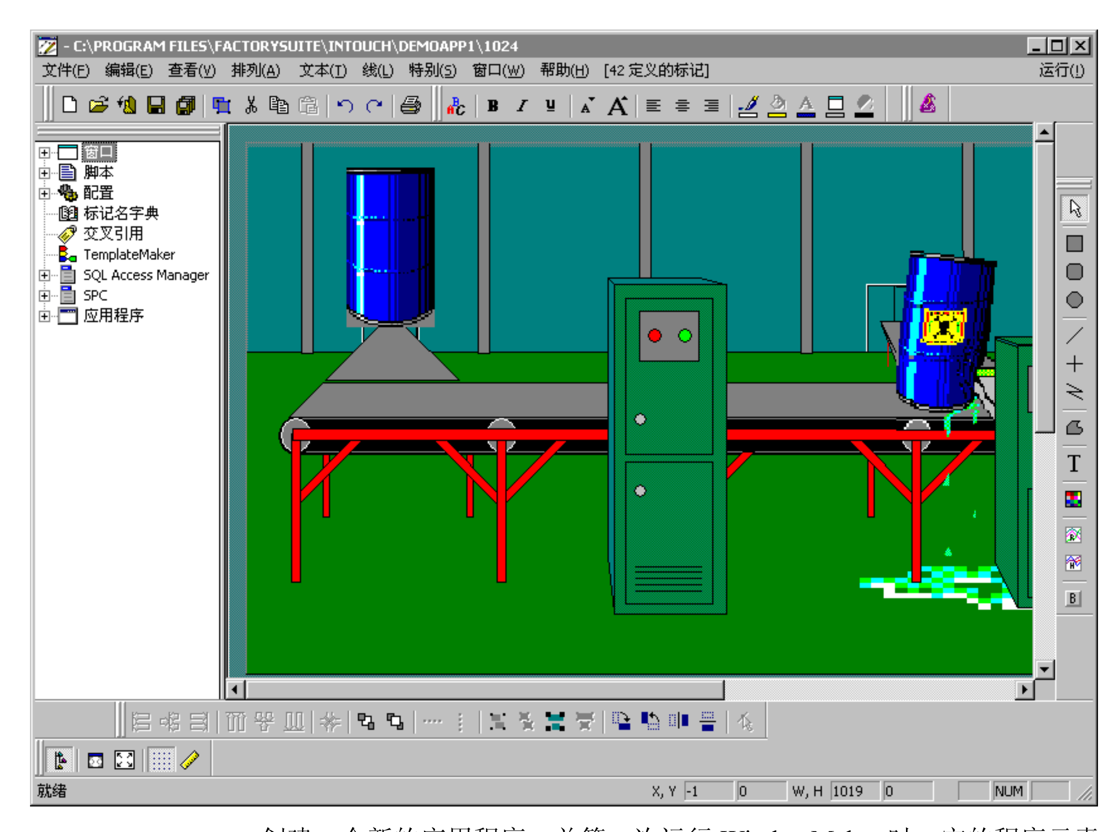
WindowMaker 图形用户界面

WindowMaker 支持 Windows 2000 和 Windows NT 操作系统的图形用户界面 （graphic user interface，简称 GUI）标准，包括右击鼠标支持、浮动和固定工具栏、下拉菜单、上下文相关帮助等。

WindowMaker 开发环境是可以配置的。在缺省条件下，最初打开 WindowMaker 时，绝大部分可用的元素均会自动显示，包括所有的工具栏、“应用程序浏览器”以及状态栏。不过，在 WindowMaker 窗口中，您可以显示或隐藏任何程序元素，也可以将工具栏和 “应用程序浏览器”移到所需的任何位置。您还可以显示可选标尺，打开或关闭窗口上的可视网格。

如需有关移动工具栏的详细信息，请参阅 “使用浮动 / 固定工具栏”。

下图说明了 WindowMaker 开发环境中的各种元素：



创建一个新的应用程序，并第一次运行 WindowMaker 时，它的程序元素会自动出现在缺省配置中，具体如上图所示。

许多工具只有在打开窗口并且在窗口中选定对象时才会被激活。某个工具未被激活时，它的功能对窗口的当前状态或所选对象将不适用。

关闭 WindowMaker 时，浮动或固定工具栏的位置和大小、“应用程序浏览器”以及 WindowMaker 窗口尺寸等首选项均会保存起来。以后再运行 WindowMaker 时，将保留上次使用的大小和位置。

应用程序浏览器

WindowMaker 的 “应用程序浏览器”是应用程序的分层结构图形化视图。它显示应用程序中已配置的项目，并让您可以轻易访问这些项目。它还可让您快速访问 WindowMaker 的许多最常用命令和功能。

备注 您可以配置 “应用程序浏览器”，以启动任何其它 FactorySuite 程序 或 Windows 程序。这项强大的功能使您能够在 HMI 配置、I/O 服务器配置和控件配置之间快速切换。

不要将 WindowViewer (VIEW.EXE) 添加到 “应用程序浏览器”中。如果添加了 WindowViewer，则在 WindowMaker 中创建的新窗口可能无法与 WindowViewer 中的窗口同步。启动 WindowViewer 的正确方法是执行**文件**菜单上的 **WindowViewer** 命令，或者单击 WindowMaker 工具栏中的**运行时**快速开关。

与所有 WindowMaker 的工具栏一样，“应用程序浏览器”可以 “固定”到 WindowMaker 窗口的任意边缘，或 “漂浮”在 WindowMaker 窗口中的任何地方。

将 “应用程序浏览器”固定到 WindowMaker 窗口的任意边缘时，它会相应自动调整大小，并在必要时显示滚动条。将 “应用程序浏览器”漂浮在 WindowMaker 窗口中时，它会显示标题栏。与所有 WindowMaker 工具栏一样，“应用程序浏览器漂浮”显示时，您可以改变其大小。

如需有关固定 / 浮动 “应用程序浏览器”的详细信息，请参阅 “使用浮动 / 固定工具栏”。

如果右击 “应用程序浏览器”的标题栏，则会出现下列菜单。



如需有关此菜单的详细信息，请参阅 “弹出式菜单”。

如需有关 “应用程序浏览器”中右击菜单功能的详细信息，请参阅 “浏览应用程序浏览器”。

要显示 / 隐藏应用程序浏览器

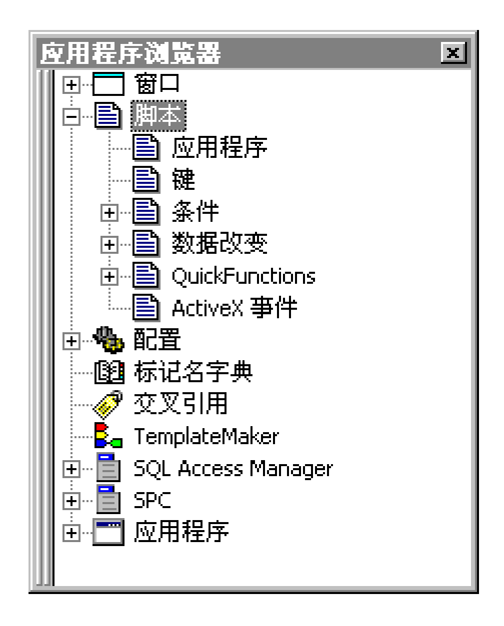
1. 在**查看**菜单上，单击**应用程序浏览器**。（初次启动 WindowMaker 时，缺省条件下会显示 “应用程序浏览器”）。
2. 重复步骤 1 以关闭 “应用程序浏览器”。

提示 要快速隐藏 “应用程序浏览器”，请单击**视图**工具栏上的 “应用程序浏览器”工具。“应用程序浏览器”漂浮在 WindowMaker 窗口中时，要快速隐藏 “应用程序浏览器”，请单击标题栏上的 ☒ 按扭，或右击标题栏，然后单击菜单上的**隐藏**。再次显示 “应用程序浏览器”时，它会以上一次在窗口中的大小和位置重新显示出来。

浏览应用程序浏览器

您可以展开或折叠在 “应用程序浏览器”分层结构图形化视图中列出的各个组。例如，如果双击某个组，则视图将展开并显示该组的成员。如果双击某个成员，则将打开该成员。例如，在**窗口**组中，如果双击某个成员窗口名，则该窗口将会打开。如果双击**标记名字典**，则会出现**标记名字典**对话框，等等。

提示 所有包含成员的组前面都会显示一个 ⊞。您可以单击 ⊞ 来快速展开该组并浏览其成员。与此类似，您也可以单击 ⊟ 来折叠该组并隐藏其成员。例如：



下节将简要介绍在执行所述动作时，“应用程序浏览器”中列出的每个组的行为：

窗口	
双击或单击 ⊞	展开视图以显示应用程序中现有的全部窗口的名称。
双击或单击 ⊟	折叠视图。
右击	打开一个弹出式菜单：
新建	打开**窗口属性**对话框。

窗口名	
双击	打开窗口。
右击	弹出一个菜单，供您从中选择可应用于所选窗口的各条命令。 撤消 选择 粘贴(P) 选择全部(A) 保存窗口(S) 关闭窗口(C) 删除窗口(D) 另存窗口为(V)... 窗口脚本(R)... QuickFunctions(Q)... 窗口属性(W)...

脚本	
双击或单击 ⊞	展开视图以显示所有的 InTouch QuickScript 类型。
双击或单击 ⊟	折叠视图。

应用程序	
双击	在 “应用程序脚本编辑器” 中打开 “应用程序脚本”
右击	此时会出现**打开**按钮。单击该按钮可以打开 “应用程序脚本编辑器”。

键	
单击 ⊞	展开视图以显示应用程序中的所有 “键脚本”。（它们按所指定的 “键” 列出）。
单击 ⊟	折叠视图。
双击	打开 “键脚本” 编辑器。（如果存在脚本，则会打开最近编辑的脚本）。
右击	出现一个弹出式菜单：
新建	打开 “键脚本” 编辑器。（如果存在脚本，则会打开最近编辑的脚本）。
打开	打开最近编辑过的脚本。

键脚本名	
双击	在键脚本编辑器中打开脚本。
右击	出现一个弹出式菜单：
新建	打开 “键脚本”编辑器。（右击的脚本会打开）。
编辑	打开选定的脚本。
删除	删除选定的脚本。

条件	
单击 ⊞	展开视图以显示应用程序中的所有“条件脚本”。
单击 ⊟	折叠视图。
双击	打开 “条件脚本”编辑器。（如果存在脚本，则会打开最近编辑的脚本）。
右击	出现一个弹出式菜单：
新建	打开 “条件脚本”编辑器。（如果存在脚本，则会打开最近编辑的脚本）。
打开	打开最近编辑过的脚本。
列出依据	打开子菜单：
名称	按其条件列出脚本
描述	按其条件列出脚本

条件脚本名	
双击	在 “条件脚本”编辑器中打开脚本。
右击	出现一个弹出式菜单：
新建	打开 “条件脚本”编辑器。（右击的脚本会打开）。
编辑	打开选定的脚本。
删除	删除选定的脚本。

数据改变	
单击 ⊞	展开视图以显示应用程序中所有的“数据改变”脚本。（它们会按所指定的标记名列出）。
单击 ⊟	折叠视图。
双击	打开“数据改变脚本”编辑器。（如果存在脚本，则会打开最近编辑的脚本）。
右击	出现一个弹出式菜单：
新建	打开“数据改变脚本”编辑器。（如果存在脚本，则会打开最近编辑的脚本）。
打开	打开最近编辑过的脚本。

数据改变脚本	
双击	在“数据改变脚本”编辑器中打开脚本。
右击	出现一个弹出式菜单：
新建	打开“数据改变脚本”编辑器。（右击的脚本会打开）。
编辑	打开选定的脚本。
删除	删除选定的脚本。

QuickFunction	
单击 ⊞	展开视图以显示应用程序中所有的 QuickFunction。（它们会按函数列出）。
单击 ⊟	折叠视图。
双击	打开 QuickFunction 编辑器。（如果存在 QuickFunction，则打开最近编辑过的 QuickFunction）。
右击	出现一个弹出式菜单：
新建	打开 QuickFunction 编辑器。（如果存在 QuickFunction，则打开最近编辑过的 QuickFunction）。
打开	打开最近编辑过的 QuickFunction。

QuickFunction 名称	
双击	在 QuickFunction 编辑器中打开 QuickFunction。
右击	出现一个弹出式菜单：
新建	打开 QuickFunction 编辑器。（右击的脚本会打开）。
编辑	打开选定的 QuickFunction。
删除	删除选定的 QuickFunction。

ActiveX 事件	
单击 ⊞	展开视图以显示应用程序中所有的“ActiveX 事件脚本”。（它们会按事件列出）。
单击 ⊟	折叠视图。

ActiveX 事件脚本名	
双击	在 “ActiveX 事件脚本”编辑器中打开脚本。
右击	出现一个弹出式菜单：
编辑	打开选定的脚本。
删除	删除选定的脚本。

配置	
双击或单击 ⊞	展开视图以显示 WindowMaker 的许多配置命令以及**向导 /ActiveX** 安装命令。
双击或单击 ⊟	折叠视图。

配置项目名	
双击	打开相应的对话框。
右击	出现一个 “打开”按钮。单击该按钮可打开项目对应的对话框。

标记名字典	
双击	打开**标记名字典**对话框，显示最近修改过的标记名的定义。否则，显示缺省的 **$AccessLevel** 系统标记名。
右击	出现一个**打开**按钮。单击该按钮可以打开**标记名字典**对话框，显示最近修改过的标记名的定义。否则，显示缺省的 **$AccessLevel** 系统标记名。

交叉引用	
双击	打开**交叉引用**实用程序。
右击	出现一个**打开**按钮。单击该按钮可以打开**交叉引用**实用程序。

TemplateMaker	
双击	打开 SuperTag **TemplateMaker** 实用程序。
右击	出现一个**打开**按钮。单击该按钮可以打开 SuperTag **TemplateMaker** 实用程序。

附加程序	
双击或单击 ⊞	展开视图以显示附加程序的配置命令。
双击或单击 ⊟	折叠视图。

程序名	
双击	打开相应的对话框。
右击	出现一个**打开**按钮。单击该按钮可以打开该命令对应的对话框。
备注 附加程序必须安装才会出现在 “应用程序浏览器”中。	

应用程序	
双击或单击 ⊞	展开视图以显示可从 WindowMaker 启动的所有其它应用程序。
双击或单击 ⊟	折叠视图。
右击	出现**新建**按钮。单击该按钮可以将一个应用程序添加到 “应用程序浏览器”中。

应用程序名	
双击	在不退出 WindowMaker 的情况下启动应用程序。
右击	出现一个弹出式菜单：
新建	打开一个空的 “应用程序属性”对话框。您可以使用此对话框来添加新应用程序。
运行	将会启动应用程序。
删除	会从 “应用程序浏览器”中删除应用程序。
属性	打开所选应用程序的 “应用程序属性”对话框。

将应用程序添加到应用程序浏览器中

WindowMaker “应用程序浏览器”最强大的功能之一是，从 WindowMaker 启动其它 FactorySuite 及第三方 Windows 应用程序。

例如，您可以在开发应用程序的同时，运行 “I/O 服务器”程序并对它进行配置。您可以启动常用的第三方 Windows 程序，例如 Windows “记事本”、“写字板”、Microsoft Excel、Microsoft Word 以及 Microsoft Paint 等。

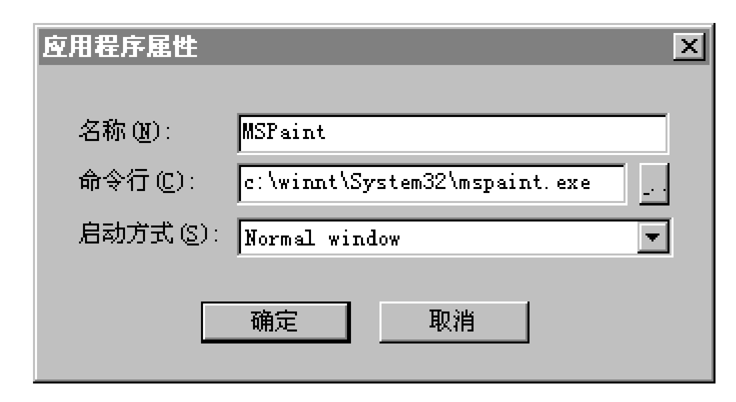
提示 InTouch 附加程序 （SQL Access、 SPC Pro 和 Recipe Manager）在安装时会自动添加到 “应用程序浏览器”中。

注意！ 不要将 WindowViewer (VIEW.EXE) 添加到 “应用程序浏览器”中。如果添加了 WindowViewer，则在 WindowMaker 中创建的新窗口可能无法与 WindowViewer 中的窗口同步。启动 WindowViewer 的正确方法是执行**文件**菜单上的 **WindowViewer** 命令，或者单击 WindowMaker 工具栏中的**运行时**快速开关。

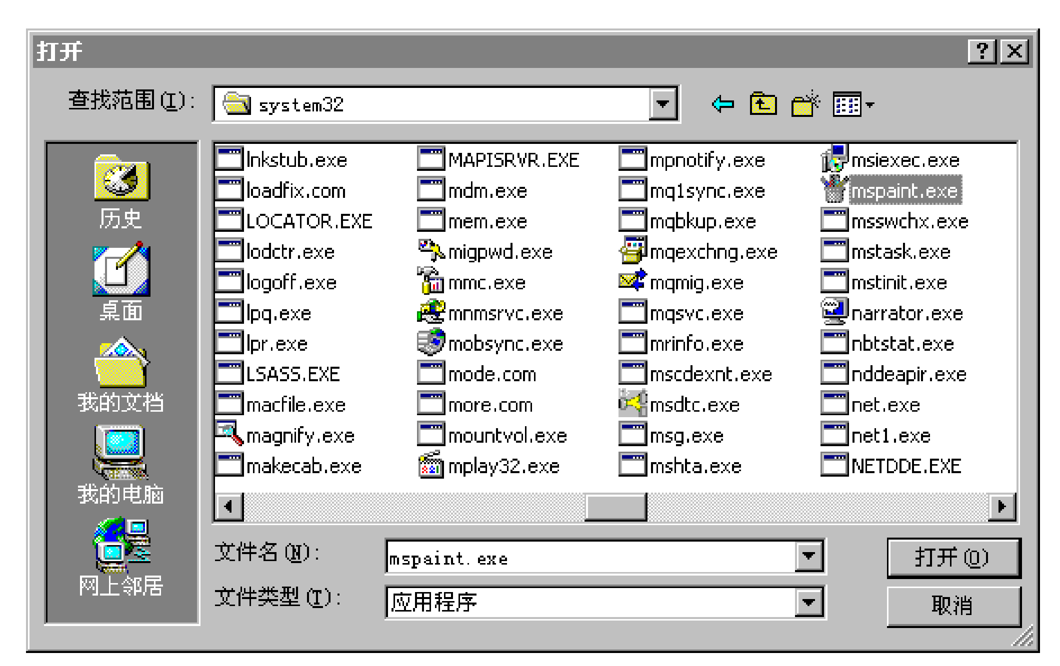
您也可以配置 “应用程序浏览器”，以便在程序中打开特定的文档或电子表格。例如，如果选择了某个特定的 Microsoft Word 文档或 Microsoft Excel 电子表格，则在 “应用程序浏览器”中双击该应用程序的图标时，应用程序会启动并自动显示所选的文档或电子表格。这些文档会显示原先创建它们的应用程序或配置为关联应用程序的 .exe 的图标。

要将应用程序添加到 “应用程序浏览器”中

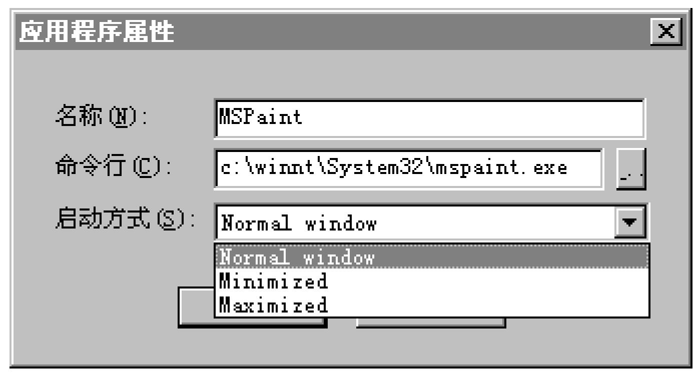
1. 显示 “应用程序浏览器”。
2. 右击**应用程序**。此时会出现一个**新建**按钮。
3. 单击**新建**。此时会出现**应用程序属性**对话框。



4. 在**名称**框中，输入要在 “应用程序浏览器”中给该应用程序显示的名称。
5. 在**命令行**框中，输入该应用程序的完整路径，或单击省略号按钮。此时会出现**打开**对话框。



6. 找到应用程序，然后单击**打开**。此时会再次出现**应用程序属性**对话框。



提示 您可以在**命令行**字段中，为该应用程序添加可选的命令行参数。

---

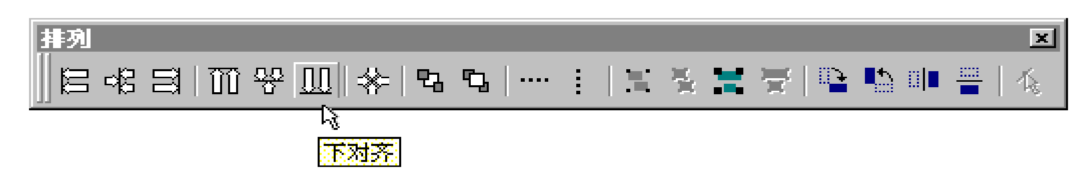
7. 单击**启动方式**箭头，选择从 WindowMaker 运行该应用程序的方式。
8. 单击**确定**。

此时该应用程序便添加到“应用程序浏览器”的**应用程序**下。现在，您可以随时从 WindowMaker 中运行该应用程序。

WindowMaker 工具栏

WindowMaker 工具栏上的工具按公共功能进行分组。例如，**排列**工具栏包含了可用于快速应用**排列**菜单上大部分命令的工具。

如果将光标置于某个工具上，则会出现一个工具提示框，显示该工具的名称。例如：



使用浮动 / 固定工具栏

WindowMaker 工具栏具有“浮动和固定”功能。这意味着您可以将任何工具栏从其缺省的“固定”位置移到 WindowMaker 窗口的任意边缘，或将其放到 WindowMaker 窗口顶部的工具栏区域。您也可以从位于窗口边缘的固定位置移动固定工具栏，让它浮动在窗口中。浮动工具栏带有标题栏，可供您更改其大小。

提示 与其它工具栏一样，“应用程序浏览器”也可以固定或浮动在窗口中的任何位置，并且其大小也可以改变。

如需有关 “应用程序浏览器”的详细信息，请参阅 “应用程序浏览器”。

要改变固定工具栏在窗口中的位置

1. 单击工具栏上的 “灰色条”，或单击固定工具栏的空白区。
2. 按住鼠标左键不放，将工具栏拖离窗口边缘、工具栏区域或 WindowMaker 窗口的任意边缘。
3. 将工具栏移动到窗口另一个边缘或工具栏区域中的新位置。

> **提示** 如果将水平固定工具栏移到 WindowMaker 窗口的左边或右边，则它在固定时会自动变成缺省的垂直形状。与此类似，如果将垂直工具栏移到窗口的顶部或底部，则它在固定时会变成缺省的水平形状。

4. 释放鼠标。此时该工具栏便固定到新的位置。

> **备注** 工具栏固定之后，您不能改变其大小，也不能访问其右击菜单。

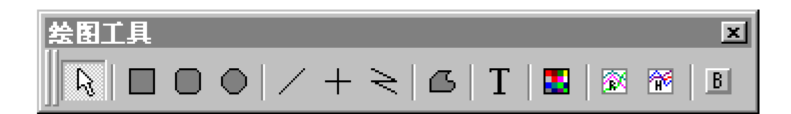
要显示 / 隐藏固定工具栏

1. 在**查看**菜单上，选择工具栏的名称。（初次启动 WindowMaker 时，缺省条件下会显示所有的工具栏）。
2. 重复步骤 1 以切换显示 / 隐藏。

> **提示** 再次显示被隐藏的固定工具栏时，它会再次出现在窗口中上次的固定位置上。

要浮动固定工具栏

1. 单击工具栏上的 “灰色条”，或单击固定工具栏的空白区。
2. 按住鼠标按钮不放，将工具栏从固定位置拖到 WindowMaker 窗口中的新位置。
3. 释放鼠标按钮。工具栏将出现并显示如下：



提示 您可以改变浮动工具栏的大小。

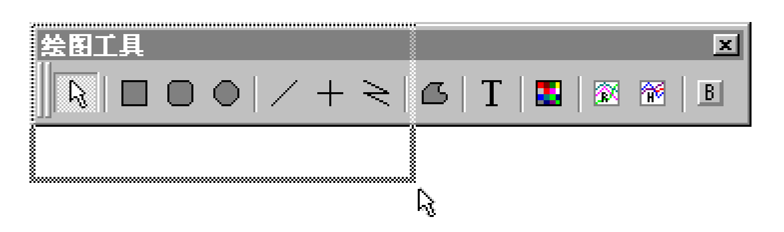
如果将浮动工具栏固定到 WindowMaker 窗口的左边或右边，则它在固定时会自动变成缺省的垂直形状。与此类似，如果将工具栏移到 WindowMaker 窗口的顶部或底部，它在固定时会变成缺省的水平形状。

固定大小已改变的工具栏时，也会发生这种情况。不过，在窗口中再次浮动工具栏时，它会恢复原来的浮动大小。

要改变浮动工具栏的大小

1. 将鼠标移到工具栏的任一边缘之上。此时光标将变成一个双向箭头。
2. 单击工具栏的边缘，按住鼠标按钮不放并拖动鼠标，以改变工具栏大小。

提示 移动鼠标时，会出现一个方框，指出释放鼠标时的工具栏大小。例如：



3. 工具栏达到所需的尺寸时，释放鼠标按钮。

提示 右击浮动工具栏的标题栏时，会出现一个菜单，显示可应用于该工具栏的各条命令。

如需有关此菜单的详细信息，请参阅 “弹出式菜单”。

要隐藏 / 显示浮动工具栏

1. 要隐藏浮动工具栏，请在**查看**菜单中，选择该工具栏的名称，或右击该工具栏的标题栏，然后单击菜单上的**隐藏**。

提示 要快速隐藏工具栏，请单击工具栏标题栏上的 ☒ 按钮。

2. 要显示被隐藏的浮动工具栏，请在**查看**菜单上，选择该工具栏的名称。

提示 此时该工具栏会以先前的大小重新出现在先前的位置上。

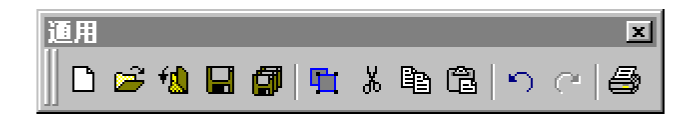
要一次性隐藏所有工具栏

1. 在**查看**菜单上，单击**隐藏全部**，或者在**视图**工具栏上单击“全部隐藏 / 恢复”工具。此时所有工具栏和“应用程序浏览器”都会隐藏起来。
2. 重复步骤 1 可以再次显示它们。

提示 在**查看**菜单上，所有显示的工具栏的名称前面都有一个选中标记。

通用工具栏

通用工具栏包括用于执行**文件**菜单上大多数窗口命令的工具，以及**编辑**菜单上的 Microsoft Windows “剪贴板”工具：



提示 右击某个打开窗口的空白区时，或者在“应用程序浏览器”中的**窗口**下右击某个窗口名时，所出现的菜单也会包含大多数这些窗口命令。

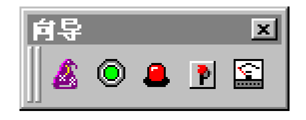
下面简要介绍每个工具：

图标名称	图标	描述
新建窗口工具		执行**文件**菜单上的**新建窗口**命令，以打开**窗口属性**对话框来创建一个新窗口。
打开窗口工具		执行**文件**菜单上的**打开窗口**命令，以打开**打开窗口**对话框，列出您可以选择打开的现有窗口的名称。
关闭窗口工具		执行**文件**菜单上**关闭窗口**命令，以打开**关闭窗口**对话框，列出您可以选择关闭的所有已打开的窗口的名称。
保存窗口工具		执行**文件**菜单上的**保存窗口**命令，以打开**保存窗口**对话框，列出自上次保存以来修改过的所有已打开窗口的名称。
保存所有工具		自动保存当前打开的且自上次保存以来已经修改过的所有窗口。此工具不会要求您逐一确认每个窗口。它会自动保存所有修改过的窗口。

图标名称	图标	描述
复制工具		执行**编辑**菜单上的**复制**命令，以复制窗口中当前选定的对象。
剪切工具		执行**编辑**菜单上的**剪切**命令，以便从窗口中剪下当前选定的对象，并将它复制到 Windows “剪贴板”上。
复制工具		执行**编辑**菜单上的**复制**命令，以复制当前选定的对象，并将它复制到 Windows “剪贴板”上。（复制的对象不会从窗口中删除）。
粘贴工具		执行**编辑**菜单上的**粘贴**命令，以粘贴任何剪下或复制到 Windows “剪贴板”的对象。（此时光标会变成粘贴模式。在窗口中单击，即可将复制或剪切的对象粘贴到窗口中）。
撤消工具		执行**编辑**菜单上的**撤消**命令，以取消最后一次应用于对象的动作或命令。
恢复工具		执行**编辑**菜单上的**恢复**命令，以取消最后一次应用于对象的撤消动作或命令。缺省条件下，撤消 / 恢复的层数设置为 10。您可以在 **WindowMaker 属性**对话框中将**撤消的级别**增加至 25。要访问此对话框，请在 “应用程序浏览器”的**配置**下，双击 **WindowMaker**，或者在**特别**菜单下，指向**配置**，然后单击子菜单上的 **WindowMaker**。
打印工具		执行**文件**菜单上的**打印**命令，以打开 **WindowMaker 打印输出**对话框，用于打印数据库和窗口信息以及 QuickScript。

向导 /ActiveX 工具栏

在缺省条件下，**向导 /ActiveX** 工具栏只包含用于访问向导**选择对话框**的向导工具。不过，您可以将任何已安装的向导或 ActiveX 控件添加到该工具栏中。例如：



下面简要介绍向导工具：

工具	描述
	显示**向导选择**对话框，用于选择要粘贴到窗口中的向导。

格式工具栏

格式工具栏包括用于执行**文本**菜单上大多数文本对象格式命令的工具。它也包含用于访问调色板的工具，以选择线、文本、填充、窗口背景和透明对象等颜色。



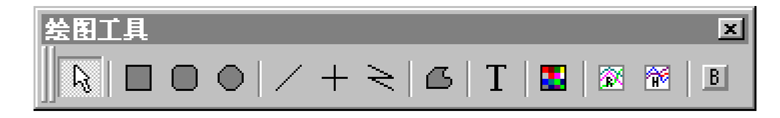
下面简要介绍每个工具：

图标名称	图标	描述
字体按钮		执行**文本**菜单上的**字体**命令，以打开**字体**对话框来选择字体、字形和大小。
粗体按钮		执行**文本**菜单上的**粗体**命令，以便对选定的单个或多个文本字符串和数值字段应用**粗体**字形。
斜体按钮		执行*文本*菜单上的**斜体**命令，以便对选定的单个或多个文本字符串和数值字段应用**斜体**字形。
下划线按钮		执行文本菜单上的**下划线**命令，以便对选定的单个或多个文本字符串和数值字段应用**下划线**字形。
字体缩小按钮		执行**文本**菜单上的**字体缩小**命令，以缩小任何字体的大小。要使用此命令，请在选择文本字符串后单击**文本**工具栏上的这个工具。
字体放大按钮		执行**文本**菜单上的**字体放大**命令，以放大任何字体的大小。要使用此命令，请在选择文本字符串后单击**文本**工具栏上的这个工具。

图标名称	图标	描述
左对齐按钮		执行**文本**菜单上的**左对齐**命令，以使选定的一个或多个文本字符串和数值字段左对齐。
居中按钮		执行**文本**菜单上的**居中**命令，以使选定的单个或多个文本字符串和数值字段居中对齐。
右对齐按钮		执行**文本**菜单上的**右对齐**命令，以使选定的单个或多个文本字符串和数值字段右对齐。
线颜色按钮		打开调色板来选择线对象或对象轮廓的颜色。
填充颜色按钮		打开调色板来选择对象的填充色。
文本颜色按钮		打开调色板来选择文本对象的颜色。
窗口颜色按钮		打开调色板来选择窗口的背景色。
位图透明颜色按钮		打开调色板来选择位图对象的透明颜色。

绘图对象工具栏

绘图对象工具栏包括用于绘制图形对象的所有工具，您既可以在窗口中绘制简单的图形对象，如长方形、椭圆、线或文本对象，也可以绘制复杂的对象，如实时趋势、历史趋势、位图和带标签的三维按钮：



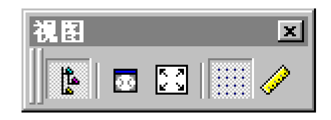
下面简要介绍每个工具：

图标名称	图标	描述
选择器模式		用于选择窗口中的对象的选择器模式。
长方形工具		用于绘制矩形或正方形的矩形工具。

图标名称	图标	描述
圆角长方形工具		用于绘制圆角长方形或圆角正方形的圆角长方形工具。
椭圆工具		用于绘制椭圆或圆的椭圆工具。
线条工具 1		用于以任意角度绘制线条的线条工具。
线条工具 2		用于绘制水平线或垂直线的线条工具。
线条工具 3		用于绘制多边线的线条工具。
多边形工具		用于绘制多边形对象的多边形工具。
字体工具		用于输入文本对象的文本工具。
位图工具		用于绘制位图容器的位图工具，可直接从 Windows “粘贴板”或下面任一文件类型粘贴位图：.BMP、.JPG、 .PCX 或 .TGA。
实时趋势工具		用于绘制实时趋势对象的实时趋势工具。
历史趋势工具		用于绘制历史趋势对象的历史趋势工具。
按钮工具		用于绘制带标签的三维按钮的按钮工具。

视图工具栏

视图工具栏包括用于执行**查看**菜单上大部分窗口命令的工具。这些命令用于控制 WindowMaker 窗口的状态。



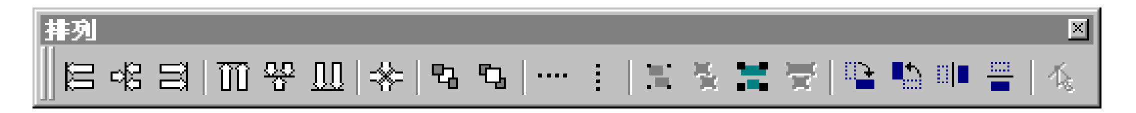
下面简要介绍每个工具：

图标名称	图标	描述
应用程序浏览器按钮		切换打开和关闭**查看**菜单上的**应用程序浏览器**命令，以显示 / 隐藏 “应用程序浏览器”。
全部隐藏按钮		切换**查看**菜单上的**全部隐藏**命令，以隐藏 / 显示所有的固定工具栏。 全部隐藏模式被激活时，WindowMaker 的总体大小保持不变。要返回正常模式，请单击浮动**视图工具栏**上的 “全部隐藏 / 恢复” 工具，或单击**查看**菜单上的**全部隐藏**。 在全部隐藏模式下，所有的浮动工具栏均保持可见，**视图工具栏**会自动浮动在 WindowMaker 顶部。如果在全部隐藏模式下固定任一浮动工具栏，则该模式会自动终止。
全屏幕按钮		切换**查看**菜单中的**全屏幕**命令，以便 WindowMaker 将显示模式从正常视图切换为全屏显示。 要返回到正常模式，请单击浮动**视图工具栏**上的 “全屏幕” 工具，或单击**查看**菜单上的**全屏幕**。 在全屏幕模式下，除了打开的窗口和浮动工具栏外，所有的 WindowMaker 程序元素均隐藏起来。**视图工具栏**会自动漂浮在 WindowMaker 顶部。 在全屏幕模式下，客户区的坐标将保持不变。例如，左上角坐标仍为 (0,0)。全屏幕模式在最大化客户区、隐藏标题栏和菜单栏并调整客户区来模仿 View 的全屏模式之后，会自动设置坐标。

图标名称	图标	描述
对齐网格按钮		切换**排列**菜单上的**对齐网格**命令，以显示 / 隐藏用于让对象对齐的可见网格。对应**排列**菜单上的**对齐网格**命令。 如果未选定 **WindowMaker 属性**对话框中的**对齐网格**选项，则此工具将不起作用。
标尺按钮		切换**查看**菜单上的**标尺**命令，以显示 / 隐藏标尺。 如需有关标尺的详细信息，请参阅"WindowMaker 标尺"。

排列工具栏

排列工具栏包括用于执行**排列**菜单上大多数对象排列命令的工具。



下面简要介绍每个工具：

按钮标题	按钮	描述
左对齐按钮		执行**排列 / 对齐**子菜单上的**左对齐**命令。以最左侧选定对象的左边缘为基准，对齐所有选定对象的左边缘。
居中对齐按钮		执行**排列 / 对齐**子菜单上的**居中对齐**命令。以选定对象组的中心线为基准，对齐所有选定对象的中心线。
右对齐按钮		执行**排列 / 对齐**子菜单上的**右对齐**命令。以最右侧选定对象的右边缘为基准，对齐所有选定对象的右边缘。
顶部对齐按钮		执行**排列 / 对齐**子菜单上的**顶部对齐**命令。以最顶端选定对象的顶边缘为基准，对齐所有选定对象的顶边缘。
垂直居中对齐按钮		执行**排列 / 对齐**子菜单上的**垂直居中对齐**命令。以对象组的中心为基准，对齐所有选定对象的中心。

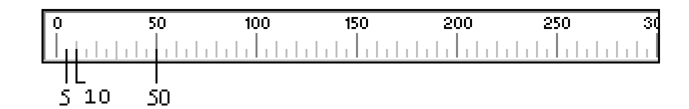
按钮标题	按钮	描述
底部对齐按钮		执行**排列 / 对齐**子菜单上的**底部对齐**命令。以最底端选定对象的底边缘为基准，对齐所有选定对象的底边缘。
中心点对齐按钮		执行**排列 / 对齐**子菜单上的**中心点对齐**命令。以选定对象组的中心点为基准，对齐所有选定对象组的中心点。
置后按钮		执行**排列**菜单上的**置后**命令，将所有选定对象置于所有未选定对象之后。
置前按钮		执行**排列**菜单上的**置前**命令，将所有选定对象置于所有未选定对象之前。
水平间距按钮		执行**排列**菜单上的**水平间距**命令，使所有选定对象均匀水平分布在最左端与最右端的选定对象之间。
垂直间距按钮		执行**排列**菜单上的**垂直间距**命令，使所有选定对象均匀垂直分布在最顶端与最底端的选定对象之间。
制作符号按钮		执行**排列**菜单上的**制作符号**命令，将多个对象组合成一个称作符号的整体。
分解符号按钮		执行**排列**菜单上的**分解符号**命令，将一个符号分解成不同的组件。
制作单元按钮		执行**排列**菜单上的**制作单元**命令，将多个选定对象合并成一个被称作单元的整体。合并单元时，每个单元都将保留，所以分解合并单元时，它们将恢复为原始单元。
分解单元按钮		执行**排列**菜单上的**分解单元**命令，以分解选定的单元。合并单元时，每个单元都将保留，所以分解合并单元时，它们将恢复为原始单元。
顺时针旋转按钮		执行**排列**菜单上的**顺时针旋转**命令，顺时针旋转选定对象 90 度。

按钮标题	按钮	描述
逆时针旋转按钮		执行**排列**菜单上的**逆时针旋转**命令，逆时针旋转选定对象 90 度。
水平翻转按钮		执行**排列**菜单上的**水平翻转**命令，以水平翻转选定的对象。
垂直翻转按钮		执行**排列**菜单上的**垂直翻转**命令，以垂直翻转选定的对象。
改变多边形或多边线形状按钮		打开**编辑**菜单上的**改变对象形状**命令，以改变多边形或多边线的形状。

WindowMaker 标尺

您可以使用 WindowMaker 标尺来精确对齐窗口中的对象。

小刻度标记的间距为 5 个象素。中刻度标记的间距为 10 个象素。带数字的大刻度标记的间距为 50 个象素。例如：



标尺的 10 和 50 个象素间距的增量相当于您按住 SHIFT 或 CTRL 键并按向上、向下、向左、向右箭头键时，选定对象移动的象素距离。

例如，如果要一次将对象移动 10 个象素，请在按住 SHIFT 键的同时按相应的箭头键。要将对象一次移动 50 个象素，请在按住 CTRL 键的同时，按相应的箭头键。

提示 如果选择对象并只按箭头键，则该对象一次将移动 1 个象素。

需要精确对齐和调整位置时，这些功能将会很有用处。

如需有关箭头键的详细信息，请参阅 “用箭头键移动对象”。

要显示 / 隐藏标尺

1. 在**查看**菜单上，单击**标尺**，或单击**查看**菜单中的标尺工具。
2. 重复步骤 1 可以隐藏标尺。

WindowMaker 状态栏

在窗口中选择对象时，WindowMaker 状态栏会显示对象的左上角 X 和 Y 象素坐标，以及该对象的象素高度和宽度。例如：



选择多个对象时，状态栏会显示最左端对象的左边缘坐标 (X) 和最顶端对象的顶边缘坐标 (Y)。同时还会显示整个组的宽度和高度。例如：

就绪 X, Y 20 50 W, H 513 120 NUM

单击窗口的空白区时，状态栏会显示窗口中光标当前位置的 X 和 Y 坐标。例如：

就绪 X, Y 310 30 NUM

要显示 / 隐藏状态栏

1. 在**查看**菜单上，单击**状态栏**。
2. 重复步骤 1 可以隐藏**状态栏**。

WindowMaker 调色板

WindowMaker 调色板支持使用多达 1.67 千万种颜色。（颜色支持仅受显卡性能的限制）。缺省条件下，调色板给您提供了范围广泛的颜色选择。不过，您可以创建自定义的调色板。自定义调色板可以加载到 WindowMaker 调色板中，或者从 WindowMaker 调色板导出。

使用标准调色板

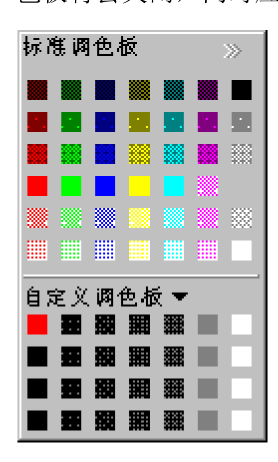
WindowMaker 调色板用于给线条、长方形、圆角长方形、椭圆、多边线、多边形及文字的静态和动态属性应用颜色。它也可用于选择窗口的背景颜色以及位图的透明颜色，以便查看位图后面的对象。

如需有关透明位图的详细信息，请参阅第 2 章 “使用 WindowMaker”。

每次单击对话框中的颜色方块，或单击某个颜色工具以便给线条、填充或文本颜色应用于所选对象时，就会出现调色板。

要使用标准调色板

1. 要选择标准颜色，请在**标准调色板**区单击要使用的颜色。（此时调色板将会关闭，同时应用您所选的颜色）。



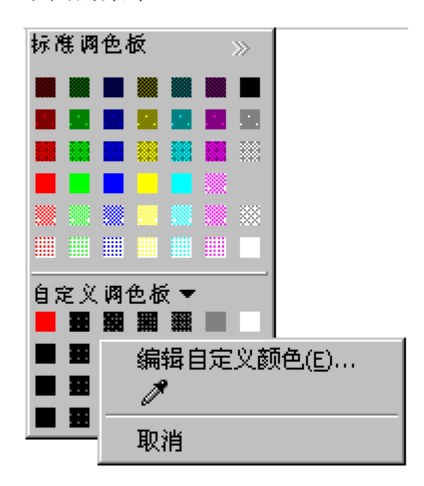
2. 要选择 InTouch 的 32 种典型颜色 （InTouch 7.0 版之前的调色板颜色）之一，请单击右角的 >>。

创建自定义调色板

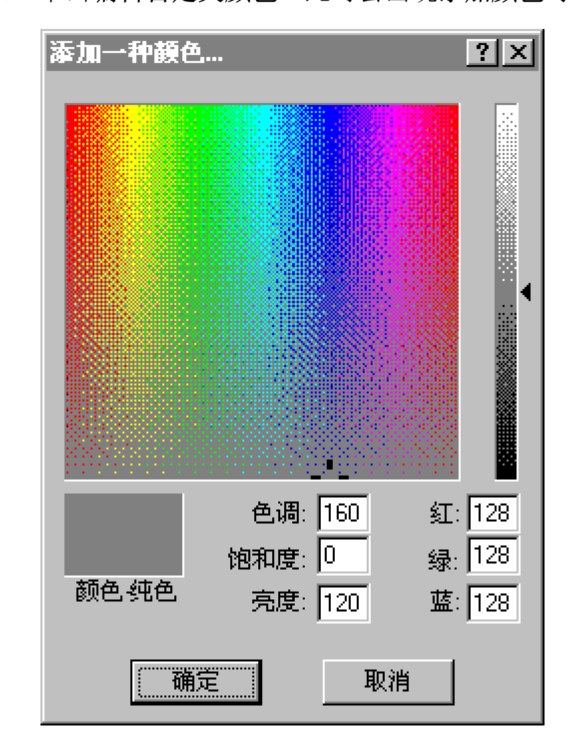
WindowMaker 调色板允许您自定义颜色并将其添加到您的调色板中，它还允许您导入其它 Windows 应用程序中创建的调色板，并将其添加到标准调色板中。您也可以将自定义调色板导出到其它 Windows 应用程序中。

要创建自定义颜色

1. 打开调色板。
2. 右击调色板底部**自定义调色板**区域中的某个空白方框。此时会出现下面的菜单。



3. 单击**编辑自定义颜色**。此时会出现**添加颜色**对话框。



4. 单击表中的任意位置，使用对话框右边的滑块来调整颜色属性。

色调、饱和度和亮度

您可以使用色调、饱和度和亮度的组合来定义任何一种颜色。色调是色盘的值，其中 0 是红色、60 是黄色、120 是绿色、180 是青色、200 是洋红色、 240 是蓝色。饱和度是指定色调中的颜色数量，最大为 240。亮度指的是颜色的亮度。如果改变其中的任意值，则红、绿、蓝的比例也会相应改变。

试用不同颜色的最简单方法是，按住鼠标不放，然后在颜色表中移动光标。

红、绿、蓝

您可以使用红、绿、蓝不同比例的组合来定义任何一种颜色。在颜色表中，您可以看到改变这些值的效果。如果改变这些值，则色度、饱和度和亮度值也将相应改变。

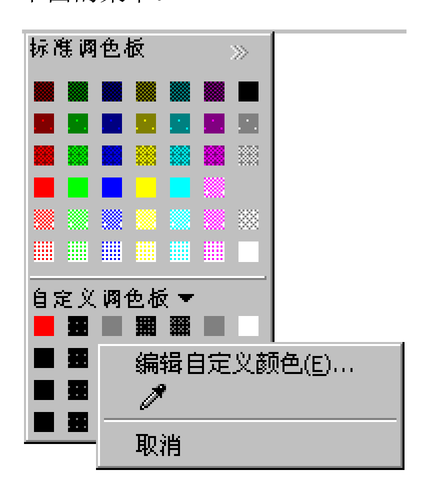
在使用**色度**、**饱和度**、**亮度**或**红**、**绿**、**蓝**比例来定义颜色时，您可以在**颜色 | 实心**框中查看该颜色，以确保所定义的颜色符合要求。

左边的**颜色**框显示指定颜色的黑色和白色数量。右边的**实心**框显示选择不含黑白色的 100% 彩色时的颜色样式。要调整颜色，请使用对话框右边的游标。要指定不含黑白色的 100% 彩色，请按 ALT + O 组合键。

5. 单击**确定**。您选定的颜色将添加到原来在调色板上单击的方块中。

要使用吸管工具来选择自定义颜色

1. 打开调色板。
2. 右击调色板底部**自定义调色板**区域中的某个空白方框。此时会出现下面的菜单。

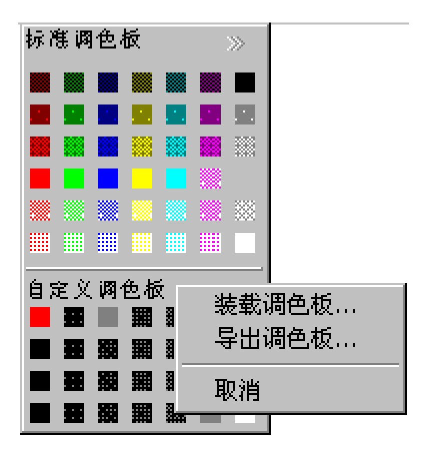


3. 单击吸管工具，然后单击要添加到调色板的**自定义调色板**区域中的颜色。您可以选择 WindowMaker 窗口中或 WindowMaker 之外任意位置的任何颜色。此功能主要在创建透明位图时使用。

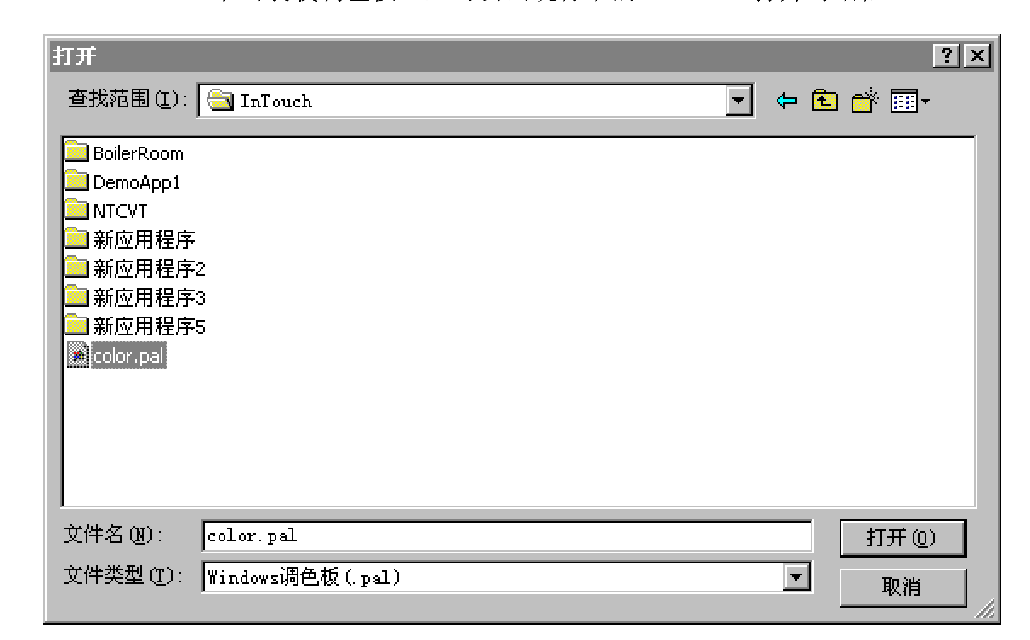
如需有关透明位图的详细信息，请参阅第 2 章 “使用 WindowMaker”。

要导入自定义调色板

1. 打开调色板。
2. 单击**自定义调色板**箭头。此时会出现下面的菜单。



3. 单击**装载调色板**。此时会出现标准的 Windows **打开**对话框。



4. 找到并选择调色板 (.PAL) 文件，然后单击**打开**，或双击文件名。调色板中包含的颜色将加载到调色板的**自定义调色板**区域中。

要导出自定义调色板

1. 打开调色板。
2. 单击**自定义调色板**箭头，然后单击菜单 （如上所示）上的**导出调色板**。
3. 此时会出现标准的 Windows **另存为**对话框。指定用以保存调色板的文件名，然后单击**保存**。

提示 调色板必须使用 .PAL 扩展名进行保存。

弹出式菜单

InTouch 支持鼠标右击功能，可显示窗口和图形对象的常用命令。弹出式菜单包含可应用于对话框、文本框、动画链接标记名或表达式框中所选文本的各条命令，在 QuickScript 窗口中也支持弹出式菜单。您不必使用标准菜单来寻找您要使用的命令，而只需在 “应用程序浏览器”中右击窗口、对象、对话框、文本框或组及其成员。

提示 要关闭右击功能，请在 INTOUCH.INI 文件中添加一行代码：**oldrightmousebehavior=1**。

要访问窗口右击菜单

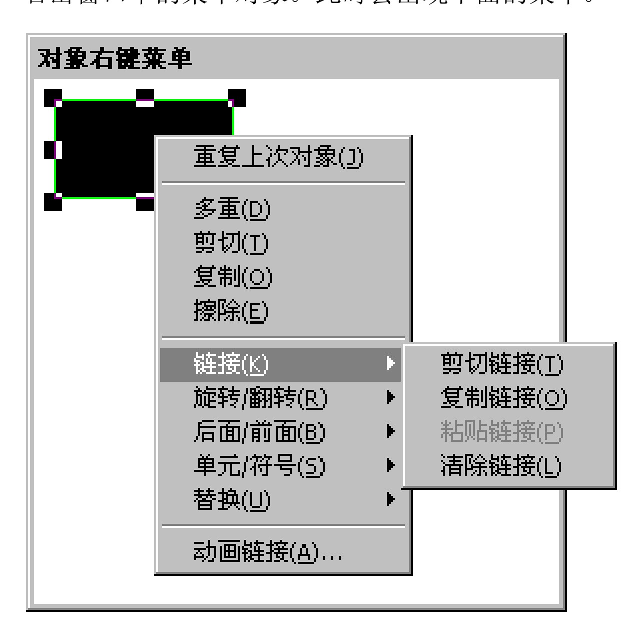
1. 右击窗口的空白区。此时会出现下面的菜单。



2. 单击菜单上要使用的命令。（不适合窗口当前状态的命令将不会被激活）。

要访问图形对象右击菜单

1. 右击窗口中的某个对象。此时会出现下面的菜单。

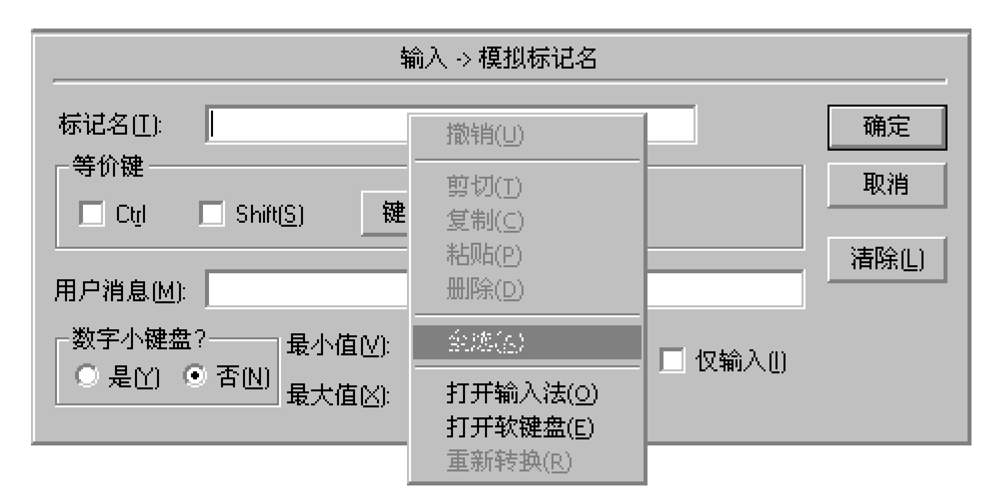


2. 单击菜单上要使用的命令。（不适合对象当前状态的命令将不会被激活）。

提示 如果某个命令存在“子菜单”，则会显示一个箭头。要选择子菜单中的命令，请先指向最初的右击菜单中的命令，然后再单击子菜单中要应用于对象的命令。

要访问对话框文本右击菜单

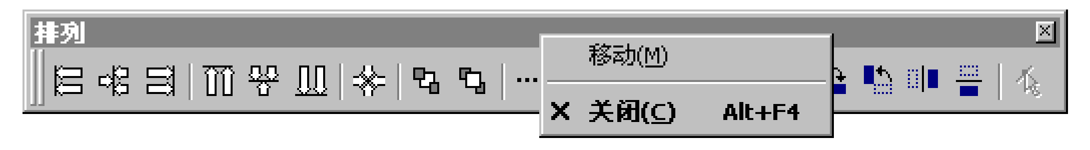
1. 在任意 WindowMaker 对话框中，右击某个文本框。此时会出现下面的菜单。



2. 单击要应用于选定文本的命令。（不适合文本当前状态的命令将不会被激活。选定部分文本或者没有选定文本时，**全部选定**命令将变成活动状态）。

要访问浮动工具栏右击菜单

1. 右击浮动工具栏的标题栏。此时会出现下面的菜单。



2. 单击菜单上要应用于工具栏的命令。

通用窗口对话框功能

使用 WindowMaker **文件**菜单上的命令打开、保存、关闭、删除或复制窗口时，您将使用的对话框都非常相似，并且具有许多通用的功能。为避免重复说明如何执行这些操作，本节将介绍这些对话框的通用功能。

右击一个打开的窗口的空白区，或单击**文件**菜单上的打开、保存、关闭、删除或另存为窗口命令时，缺省条件下，所选命令对应的对话框会出现在 “列表视图” 中。这意味着适合所选命令的所有窗口的名称会出现在一个连续的列表中。例如：



提示 窗口名数目超过缺省的列表空间时，会出现一个水平滚动条。

单击**详细资料**可以将 “列表视图” 改变为详细视图。

选择详细视图时，屏幕将以多列格式显示各个窗口及其详细资料。显示的详细资料包括在**窗口属性**对话框中输入的任何窗口注释、窗口类型，以及最后一次修改窗口的日期和时间。例如：



在详细视图中，您可以选择任何一个未打开的窗口，方法是单击它所在行的任何部分，而不只是复选框。（此时整行将辉亮显示）。再次单击选定的窗口，可以取消选择它。窗口名数目超过缺省的列表空间时，会出现一个垂直滚动条。

要按详细资料类型给列表排序，请单击该详细资料的列标题。详细视图的排序顺序如下：

列	排序顺序
名称	字母
注释	字母
类型	覆盖、替换、然后弹出
上一次修改	最早的日期 / 时间 （顶端）到最近的日期 / 时间（底端）

提示 每次单击列标题时，列表的排列顺序均会从升序变为降序，或正好相反。例如，如果列表当前以升序排列，在您单击某个列标题时，该列表将按降序重新排序选定的列。

要让列表返回缺省显示状态，请单击列标题最左端的小方框。

要调整列大小，请将光标放到每个详细资料列标题的垂直分隔线上。光标变成 "I" 形时，请单击并拖曳该标题至所需宽度。

要快速自动调节列大小，请双击该列的右垂直分隔线。

要打开选定的窗口，请单击**确定**。

要取消选择并关闭对话框，请单击**取消**。

要让对话框返回 “列表视图”，请单击 “列表”。

要选择列出的所有窗口，请单击**全部选定**。

要清除选定的所有窗口，请单击**全部清除**。

其它鼠标快捷方式

双击某个对象或符号，可以自动使用所选的对象或符号执行**动画链接**命令 （在**特别**菜单上）。

如需有关动画链接的详细信息，请参阅第 7 章 “创建动画链接”。

双击 “链接定义”对话框中的空白表达式输入字段，可以启动 “标记浏览器”，列出该应用程序 “标记名字典”中定义的所有标记名。

在 “链接定义”对话框的表达式输入字段中的句点 **(.)** 之后双击鼠标左键，将会显示**选择域名**对话框，其中包含所有标记名**点域**的全局列表。

双击动画链接标记名或表达式中的标记名，则会打开 “标记名字典”中该标记名的定义。

双击动画标记名或表达式中的 SuperTag 父模板名，会打开该 SuperTag 成员标记名的 “标记名字典”详细资料对话框。

如需有关 “标记浏览器”和标记名点域的详细信息，请参阅第 6 章 “标记名字典”。

右击一个打开的窗口的空白区、任意 WindowMaker 对话框中的文本框或图形对象，均会显示一个可菜单，其中包含您可以应用于窗口、文本或对象的各条命令。右击浮动工具栏的标题栏也将显示一个您可以应用于该工具栏的命令菜单。

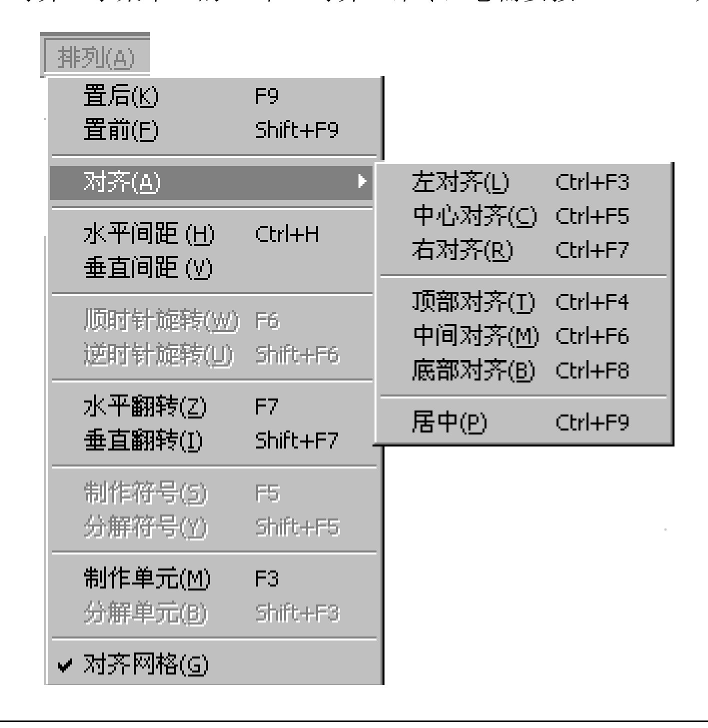
快捷方式和快捷键

InTouch 提供了许多鼠标和键盘快捷方式供您访问一些常用的功能。只要菜单命令包含键盘快捷方式，则它会显示在菜单上此命令的右边。此外，所有命令都可以用一个以 ALT 键开始的三键序列加以执行。第二个键是菜单名中带下划线的字符，第三个键是命令中带下划线的字符。

例如，您可以使用序列 alt + fn 来执行 “文件”菜单上的 “新建窗口”命令。

要执行右击菜单上的某个命令，您可以输入该命令中带下划线的字符。

要执行子菜单上的某个命令，您必须依次按下三个键。例如，要执行“对齐”子菜单上的 “中心对齐”命令，您需要按 alt + aac 或 ctrl+F5



备注 选择一个后跟 (**...**) 的菜单命令时，会出现一个对话框，您必须选择或输入更多的信息才能完成此命令。

用箭头键移动对象

在 WindowMaker 中，您可以使用上、下、左、右箭头键，将选定的对象或对象组朝所按箭头键的方向一次移动一个象素。需要精确对齐和调整位置时，此项功能会非常方便。要将对象一次移动 10 个像素，请在按住 SHIFT 键的同时按箭头键。要将对象一次移动 50 个像素，请在按住 CTRL 键的同时按箭头键。

在 WindowViewer 中使用箭头键时，系统使用一定的算法从将一个触控对象移动到另一个对象。例如，左光标箭头键向左移动的宽度与所选对象的高度相等。如果它在运动路线中与另一个触控对象部分相交，则后者将取代当前选定的对象获得焦点。如果没有遇到任何其它触控对象，则运动路线将在屏幕左边界绕回，并继续试图与另一个对象相交。如果未遇到其它对象，则原始对象仍将保持选定状态。

所有箭头键的工作方式均相同。向上和向下箭头键使用选定对象的宽度作为其运动路径的宽度。

提示 如果您的应用程序用户只使用光标键在应用程序窗口中浏览，则必须认真设计窗口中的触控对象布局，以确保其运动路径相交。

在 WindowViewer 中，也可以使用 TAB 键在窗口中的触控对象之间进行转换。（不过，无法保证跳格顺序）。

只要在设计应用程序时小心，加上理解箭头键的作用方式，您就能够创建一个不用鼠标的应用程序。

使用 WindowMaker 帮助

WindowMaker 支持上下文相关帮助，具体如下：

要获取关于打开的对话框的帮助信息，请按 F1 键。此时会出现相应的 Windows 联机帮助主题。

要在 QuickScript 编辑器中获取关于各种 QuickScript 函数的帮助信息，请单击**帮助**，或在**插入**菜单上指向**函数**，然后单击**帮助**。此时会出现一个对话框，列出所有可用的 QuickScript 函数。单击您要获取其帮助信息的函数。此时会出现该函数对应的 Windows 联机帮助主题。

要获取有关 InTouch 软件的信息，例如您使用的版本号、序列号、许可证截止日期等，请在 WindowMaker 的**帮助**菜单上，单击**关于**。此时会出现**关于 InTouch** 对话框。



单击**查看许可证**可以启动许可证查看实用程序，以获取有关 FactorySuite 许可证的信息。

如需有关“许可证查看实用程序”的详细信息，请参阅 *FactorySuite 系统管理员指南*。

第 2 章

使用 WindowMaker

通过设置 WindowMaker 和 WindowViewer 的各种属性，您可以自定义应用程序的功能和最终外观。例如，您可以指定 WindowViewer 中的可用菜单，也可以在应用程序的标题栏中显示公司名称，等等。

本章介绍如何配置 WindowMaker 和 WindowViewer、使用 WindowMaker 的窗口、编辑和排列图形对象，以及如何安装和使用向导及 ActiveX 控件。

目录

简单对象

WindowMaker 有四种基本类型的简单对象：线条，填充图案、文本和按钮。每种简单对象类型都包含影响其外观的属性。这些属性包括线条颜色、填充颜色、高度、宽度和方向等，并既可以是静态，也可以是动态的。静态属性在应用程序运行期间保持不变。动态属性与表达式值链接，因此表达式值发生的变化会导致属性发生变化。例如，对象的填充颜色可以与一个离散表达式的值链接。根据表达式的状态，在表达式为真时，填充颜色会是一种颜色；在表达式为假时，填充颜色会是另一种颜色。简单对象的大多数属性都可以设置为动态。一个对象可以有一个以上的动态属性。动态属性可以任意组合，以取得所需的结果。下列简要介绍 WindowMaker 的简单对象类型：

对象	描述
线条	根据线条类型，线条对象可以由一个或多个线段组成。颜色是线条可以设置动画效果的唯一属性。宽度和样式不能形成动画效果。它们只能简单地指定缺省属性。
填充图案	填充图案是由线条包围的闭合内部区域组成的二维对象。填充图案的例子有矩形、圆角矩形、圆、椭圆及多边形等。填充图案的属性包括：线条颜色、线条宽度、线条样式、填充颜色、颜色填充百分比、高度、宽度、位置、可见性、方向和大小。
文本	文本是单行字符串起来组成的对象。文本对象的属性包括：字体、大小、颜色、粗体、下划线、*斜体*、对齐方式、可见性和位置。
按钮	通过使用 WindowMaker “绘图对象工具栏”上的“按钮”工具，可以创建任意大小的三维按钮。按钮表面的缺省 “文本”字符串可以使用 “特别”菜单上的 “替换字符串”命令进行编辑。 按钮可以附加许多类型的链接，如动作脚本、键脚本、模拟或离散值输入或输出链接。如果按钮附加了输入或输出链接，则它的值会作为文本字符串显示在按钮上。

复杂对象

除了简单对象外， InTouch 也支持使用有着很大不同的各种复杂对象。下面简要介绍 WindowMaker 的复杂对象类型：

对象	描述
位图	“位图”工具用于将位图复制并粘贴到应用程序中。粘贴到应用程序窗口支欧，位图就可以旋转，也可以给它定义透明背景，使之浮在其它对象之上。
趋势	趋势工具有二种：一种用于创建显示实时数据的趋势，另一种用于创建显示历史数据 （可从历史记录文件中检索）的趋势。实时和历史趋势二者都可以进行配置，以显示特定时间内多个标记名的图形化表示。
符号	符号是简单对象 （线条、填充图案和文本）的组合，它被视为单个对象。符号的任何属性发生改变时，不管是 WindowMaker 中静态属性的改变，还是 WindowViewer 中动态属性的改变，都会影响该符号中所有的组件对象。 例如，如果使用两个圆和两个矩形创建一个泵符号，然后在该符号上定义一个填充颜色链接，则该填充颜色将应用于全部四个对象。与此类似，在 WindowMaker 中，选择 “填充颜色”也会改变所有组件对象的填充颜色。 对于相同的属性，符号的组件对象可以具有不同的值，条件是在将对象合并成符号之前，这些属性不同，并且在变成符号之后这些属性没有改变。符号不能包含位图、按钮、单元、报警或趋势。
单元	单元是两个或更多对象、符号或其它单元的集合，它们相互结合形成一个单一的整体。单元在各自的图形元素之间保持固定的空间关系。单元的每个组件都有它自己的链接。单元用于创建虚拟设备，如游标控制器。 要创建一个单元，请选择两个或更多对象、符号和 （或）单元，然后执行**排列**菜单上的**制作单元**命令。 将一组对象制作成单元之后，它的内部细节如颜色、大小和动画链接等将不能再改变。改变单元外观或操作的唯一方法是使用**排列**菜单上的**分解单元**命令将它 “拆开”。 在 WindowViewer 中，通过链接操作可以改变单元组件的属性。您可以对单元执行复制、拷贝、粘贴、对齐和更改间距等操作。单元的大小不能调整。您必须先分解它们，调整大小之后再重新将它们制作为一个单元。在创建与不同标记名关联的多个相似设备时，单元非常有用。

对象	描述
向导	在应用程序开发阶段，向导可以节省您大量的时间。它们易于使用且便于配置。选择向导并将它粘贴到窗口而后再双击它时，它的配置对话框将会出现，供您对它进行配置。 例如，对于 “游标” 向导，配置项目包括激活标记名、游标的最小和最大范围标签以及填充颜色等。所需的配置信息输入完毕之后，就可以开始使用该 “向导”。通过使用 “向导”，您不必再花时间去绘制对象的各个单独组件，输入对象的值范围，或给对象设置动画效果。 您可以开发出 “复杂的” 向导来提供 “后台” 型操作。这些操作包括创建完整的显示窗口、创建或转换数据库、导入 AutoCad 图形，以及配置其它应用程序，如 InTouch Recipe Manager 和 SPC 等附加程序。 **备注**熟练的 "C" 程序员可以使用 Wonderware 的 Extensibility Toolkit 来开发自定义的 “向导”。此工具包可以从经销商处购得。 如需有关 “向导” 的详细信息，请参阅 “使用向导”。

对象	描述
ActiveX 控件	WindowMaker 支持 ActiveX 控件，最简单形式的控件是与应用程序对话或在应用程序中运行的微型应用程序。WindowMaker 支持 Wonderware FactorySuite 组件中包含的所有 ActiveX 控件，例如所有的 InTrack ActiveX 控件。 WindowMaker 也支持第三方 ActiveX 控件，如 Office97 中所安装的那些控件。 安装 InTouch ActiveX 控件的方法与安装任何其它向导一样。如果需要，您可以将常用的 ActiveX 控件添加到 WindowMaker 的**向导 /ActiveX** 工具栏。 选择 ActiveX 控件并将它粘贴到窗口然后再双击它时，它的配置对话框便会出现。配置 ActiveX 控件时，您可以给它指定一个唯一的控件名，以便可以在脚本中引用它 （最初添加控件时，会创建一个缺省的名称）。 所有的 ActiveX 控件都具有与之关联的属性、方法和事件。您可以将 ActiveX 属性与相应数据类型的标记名关联起来。您可以通过 InTouch QuickScript 函数执行 ActiveX 方法。您可以将 ActiveX 事件与事件发生时执行的 “ActiveX 事件脚本”关联起来。换句话说，您可以使用 InTouch QuickScript 函数来处理控件事件、调用控件方法和控件属性。 在运行时，您在 WindowMaker 中定义的标记名和 QuickScript 将控制 ActiveX 控件的行为。

自定义开发环境

您可以设置许多属性来自定义 WindowMaker。例如，您可以自定义应用程序的标题栏文本，使之包含公司名称。您还可以设置网格的像素间距等。

要设置 WindowMaker 的属性

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowMaker**，或者在 “应用程序浏览器”中的**配置**下，双击 **WindowMaker**。此时会出现 **WindowMaker 属性**对话框，并显示**通用**属性页。

提示 在“应用程序浏览器”的**配置**下，右击 **WindowMaker**，然后单击**打开**。



提示 如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 在**标题栏文本**框中，输入要在运行时显示在应用程序标题栏中的标题。例如：

   ABC Company, Paint APP1

备注 如果使用“推广版许可证”，则不能更改标题栏。

如需有关 FactorySuite 授权方式的详细信息，请参阅 *FactorySuite 系统管理员指南*。

3. 如果要在标题栏中包括应用程序的目录路径，请选择**显示应用程序目录**。例如：

   ABC Company, Paint APP1 - C:\DEMOAPP1

4. 在**间距**框中，输入对齐网格坐标间隔的像素数。
5. 如果要在打开 WindowMaker 的“对齐网格”功能时在窗口中显示网格，请选择**显示网格**。

提示 如果不选择**显示网格**，则打开对齐网格功能时在窗口中会看不到网格。

单击**视图**工具栏上的“对齐网格”工具，或者在**排列**菜单上单击**对齐网格**，以打开或关闭对齐网格的功能。

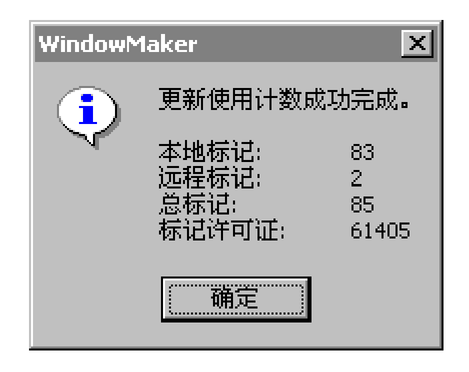
对象均会用自己的左上角对齐网格。如果选择多个对象，则会按组中第一个选定对象的左上角来对齐网格。

6. 如果要在 WindowMaker 菜单栏上显示“标记名字典”中已定义的标记名数，请选择**显示标记计数**。

提示 如果使用有限大小的“标记名字典”创建应用程序，则这会是非常有用的信息。

备注 显示标记名计数会严重影响 WindowMaker 中“标记名字典”的性能。

显示的标记名计数不包括远程标记名引用。要确定远程标记名引用的使用情况，请在**特别**菜单上执行**更新使用计数**命令。在系统更新使用计数完毕之后，会出现下列对话框。



如需有关更新使用计数的详细信息，请参阅第 6 章“标记名字典”。

7. 如果希望在启动 WindowViewer 时自动关闭 WindowMaker，请选择**转换到 WindowViewer 时关闭**。

备注 如果内存足够并且您经常在 WindowViewer 与 WindowMaker 之间切换，请不要选择此选项。

提示 选择此选项时，同时也会自动选定 **WindowViewer 属性**对话框**通用**属性页上的**关闭 WindowViewer** 选项。

8. 如果要在 WindowMaker 和 Scrapbook+ 之间复制和粘贴对象，请选择**启用 Scrapbook 菜单项**。

备注 要使用这些菜单命令，您的计算机上必须已安装 Windows Scrapbook+ 程序。此选项供旧的 InTouch 系统使用。 FactorySuite 不支持此选项， Wonderware 也不会去销售 Scrapbook+。

9. 如果希望能够去选择 “中空”对象后面的对象，请选择**透过中空对象选取**。

   提示 如果选择此选项，然后绘制四条线并将它们连成一个包围另一个对象的框架，则您可以选择框架内的对象而不必将框架置后。

10. 如果希望在 WindowMaker 与 WindowViewer 的菜单栏上显示两个程序之间的快速切换开关，请选择**启用快速切换**。

    提示 如果选择此选项，则在 WindowMaker 中，快速切换开关是屏幕右上角显示的**运行时**字样。在 WindowViewer 中，它是**开发**字样。要在两个程序之间快速切换，请单击快速切换开关。

    备注 使用快速切换开关时， WindowMaker 会在 WindowViewer 启动之前，自动保存对所有打开窗口所作的修改。

11. 在**行选择精度**框中，输入光标可以远离行但仍能选定它的像素数。

    提示 在大多数情况下，使用缺省设置 4 就足够了。

12. 在**撤消级别**框中，输入要保存的撤消 / 恢复级数。您最多可以选定 25 级。如果输入 0，则会关闭撤消 / 恢复功能。

    一个级别表示一个动作。创建新窗口或打开现有的窗口时，“撤消”与 “恢复”堆栈均为空。保存窗口时，这两个堆栈均会被清空。

    如需有关撤消与恢复的详细信息，请参阅 “撤消对象编辑”。

13. 单击**确定**。

备注 修改上述任意设置之后，必须重新启动 WindowMaker 才能使所作的更改生效。

使用 WindowMaker 窗口

InTouch 应用程序更像是由用于存放图形和文本对象的一系列窗口组成。在 WindowMaker 中创建新窗口时，程序会要求您定义该窗口的特定属性，例如背景色、标题和屏幕位置等。您也可以创建 QuickScript，根据窗口是打开、显示还是关闭而执行。

本节包含创建、打开、保存、关闭、删除和复制窗口时需要遵循的操作程序。

提示 **通用工具栏**包含可用于快速应用**文件**菜单上大多数窗口命令的工具。

要快速访问可应用于窗口的各种命令，请右击打开窗口的空白区，然后单击右击菜单上适当的命令。例如：



创建新窗口

要创建一个新窗口

1. 在**文件**菜单上，单击**新建窗口**，或者单击**通用工具栏**上的**新建窗口**工具。此时会出现**窗口属性**对话框。

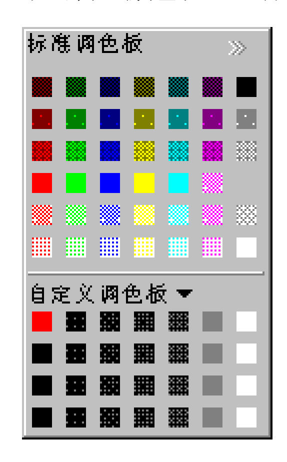
提示 要快速创建一个新窗口，请在“应用程序浏览器”中右击**窗口**，然后单击**新建**。



备注 缺省条件下，此对话框中的设置会反映先前创建的窗口的属性；或者，如果您是在 WindowMaker 中有窗口打开的情况下选择此命令，则这些设置会反映活动窗口的属性。如果活动窗口附带了 “窗口” 脚本，则会出现一个消息框，询问是否要将这些窗口脚本复制到新窗口中。

提示 如果右击任意对话框中的任何文本输入框，则会出现一个菜单，显示可应用于所选文本的命令。

2. 在**名称**框中，输入要在新窗口的标题栏中显示的名称。名称最长可达 32 个字符。它可以包含嵌套空格、标点符号和键盘上的任何其它字符，但引号 (") 除外。
3. 在**注释**框中，输入要与窗口关联的任何其它注释 （可选）。此信息仅出于编制文档之目的，应用程序不会去使用它。
4. 单击**窗口颜色**框，选择窗口的背景颜色。此时会出现调色板：



提示 如果不必改变，请单击当前颜色选项或按 ESC 键关闭调色板。

如需有关使用调色板的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

5. 单击要用作窗口背景的颜色。
6. 选择要使用的**窗口类型**。窗口类型共有三种：

窗口	描述
替换	此类型的窗口出现在屏幕上时，它会自动关闭与它相交的任意窗口，包括弹出型和其它替换型窗口。

窗口	描述
覆盖	出现在当前显示窗口的上方，可以比它所覆盖的窗口更大。覆盖窗口关闭之后，所有被覆盖的窗口会重新显示出来。单击被覆盖窗口的任意可见部分，便可将该窗口置前并激活。
弹出	类似于覆盖窗口，其区别在于弹出式窗口总是位于所有其它打开的窗口上方 （即使单击了另一个窗口）。弹出式窗口通常需要用户作出响应才能移除。

提示 您可以通过再次打开它的**窗口属性**对话框来改变窗口的类型。方法有三种：

打开窗口，然后在**窗口**菜单上，单击**窗口属性**。

在 “应用程序浏览器”中的**窗口**下，右击窗口名，然后单击**属性**。（如果执行该命令时没有窗口打开，则它会自动在对话框的后面打开）。

打开窗口，右击窗口的空白区，然后单击**窗口属性**。

7. 选择窗口的**框式样**。式样有三种：

框架	描述
单一	带标题栏、**大小控制**且具有三维边框的窗口。
加倍	不带标题栏、“大小控制”且具有三维边框的窗口，其大小也因此无法调整。
无	没有边框、不带 “大小控制”的窗口，其大小也因此无法调整。（如果带“大小控制”，则它会变成一个可调整大小且具有三维边框的窗口）。

8. 如果希望窗口带标题栏，请选择**标题栏**。通过在标题栏上单击并拖动鼠标，标题栏也可以用于移动窗口。

备注 如果窗口有标题栏，则不能给**框式样**选择**加倍**或**无**。

9. 如果希望用户能够在 WindowViewer 中改变窗口大小，请选择**大小控制**。
10. 在**位置大小**组中，输入每个窗口坐标的像素位置：

位置	描述
X 轴位置	WindowMaker 设计区的左边缘与要定义的窗口左边缘之间的像素数目。
Y 轴位置	WindowMaker 设计区的顶边缘与要定义的窗口顶边缘之间的像素数目。

位置	描述
窗口宽度	窗口宽度的像素值。
窗口高度	窗口高度的像素值。

备注 Windows 根据显示器来限制窗口的最小高度值。例如，对于标准的 VGA 显示器，最小值为 26 个像素。 Windows 根据显示器来限制窗口的最小宽度值。例如，对于标准的 VGA 显示器，最小值为 102 个像素。

备注 缺省条件下，这些输入框的值将设置为先前创建的窗口的尺寸。如果在 WindowMaker 中通过窗口边框手动调整窗口大小，则会自动修改这些值。

11. 单击**脚本**以访问**窗口脚本**编辑器。您可以给窗口应用下列三种类型的脚本：

脚本	描述
显示时	在窗口最初显示时执行一次。
显示期间	在窗口显示期间按指定的频率连续执行。
隐藏时	在窗口隐藏时执行一次。

备注 如果将 “窗口脚本”附加到活动窗口，然后创建一个新窗口，则活动窗口的脚本可以复制到新窗口中。此时会出现一个消息对话框，询问是否要复制窗口脚本。

提示 如果后来您决定需要将脚本附加到打开的窗口，请右击打开窗口的空白区，然后单击**窗口脚本**。如果窗口没有打开，请在 “应用程序浏览器”中双击**窗口**，以显示所有的窗口名。右击窗口名，然后单击**窗口脚本**。

如需有关创建窗口脚本的详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

创建一个隐藏标题栏和菜单栏的窗口

WindowMaker 的设计区是窗口框架内标题栏和菜单栏下方的整个区域。在 WindowViewer 中，设计区会变成显示区。指定的 X=0 和 Y=0 位置始终正好在标题栏和菜单栏下的左上角。标题栏和菜单栏均为 19 个像素高，并且位于设计区上方。例如，如果 WindowMaker 被最大化，而您使用的是 1024x768 的显示器，则可见设计区为 1024x730 （768 像素减去标题栏和菜单栏各占的 19 像素，得出可见设计区像素值 730）。如果 WindowViewer 配置为显示标题栏和菜单栏，则 WindowViewer 的显示区与 WindowMaker 完全一样，包括标题和菜单栏下方的整个屏幕。

为充分利用被标题栏和菜单栏占用的空间，您可以设计一个隐藏标题栏和菜单栏的应用程序。隐藏标题栏和菜单栏时，窗口的左上角会对应屏幕上的一个新位置。这就增加了可见显示区，为您提供了更多的显示空间。如果按照这种方式配置 WindowViewer，则所有的窗口会自动上移，窗口底部会出现一个缺口。要填充此缺口，您需要将窗口的 Y 坐标设置为负值来增加窗口高度。结果是，在 WindowMaker 中，窗口位于标题栏和菜单栏的下方，而在 WindowViewer 中，窗口位于标题栏和菜单栏的上方。

您可以利用此项技术在 WindowViewer 中用一个弹出式窗口来覆盖标题栏和菜单栏。您也可以创建一个触动链接按钮或 QuickScript，在需要让用户看见标题和菜单栏时，隐藏这个弹出式窗口。此外，通过应用安全设置和口令，您还可以限制特定用户无法隐藏窗口和访问菜单，包括退出 WindowViewer 的能力。

备注 推广版 InTouch 许可证不允许隐藏标题栏。因此，如果应用程序是在推广版 InTouch 许可证下开发的，并且 WindowViewer 的配置是在标准运行时许可证下查看应用程序时选定**隐藏标题栏**，则标题栏将根据配置而被隐藏，并且应用程序中的所有窗口将向上移动。

打开窗口

在开发应用程序期间，只要计算机内存足够，您可以打开任意多个窗口。

要打开窗口

1. 在**文件**菜单上，单击**打开窗口**，或者单击**通用工具栏**上的 “打开窗口” 工具。此时会出现**打开窗口**对话框，列出应用程序中所有的窗口名。

   提示 要快速打开一个窗口，请在 “应用程序浏览器” 中，双击**窗口**以打开应用程序中所有窗口名的列表，然后双击所需的窗口名。您也可以右击窗口名，然后单击**打开**。

2. 单击要打开的窗口名旁边的复选框。

   提示 缺省条件下，当前打开的所有窗口均已选定。

3. 单击**确定**以关闭该对话框，并打开选定的窗口。

保存窗口

创建窗口之后，您需要在关闭它或退出应用程序之前保存它。所有与窗口关联的图形、 QuickScript 和属性等均会得以保存。

要保存窗口

1. 在**文件**菜单上，单击**保存窗口**，或者单击**通用工具栏**上的 “保存窗口” 工具。此时会出现**保存窗口**对话框，列出所有需要保存的窗口名。

提示 要快速保存一个窗口，请在 “应用程序浏览器” 中，右击该窗口名，然后单击**保存**。您也可以右击窗口的任意空白区，然后单击**保存窗口**。

要快速保存当前打开的所有窗口，请单击**通用工具栏**上的**保存所有窗口**工具，或者单击**文件**菜单上的**保存所有**命令。

2. 单击要保存的窗口名旁边的复选框。
3. 单击**确定**以关闭对话框并保存选定的窗口。

关闭窗口

如果试图关闭自上次保存以来已经修改过的窗口，则程序会在 WindowMaker 关闭窗口之前，提示您保存所作的修改。

要关闭窗口

1. 在**文件**菜单上，单击**关闭窗口**，或者单击**通用工具栏**上的 “关闭窗口” 工具。此时会出现**要关闭的窗口**对话框，列出当前打开的所有窗口名。

提示 要快速关闭某个窗口，请在 “应用程序浏览器” 中，右击窗口名，然后单击**关闭**。您也可以右击窗口的任意空白区，然后单击**关闭窗口**。

2. 单击要关闭的窗口名旁边的复选框。
3. 单击**确定**以关闭对话框并关闭选定的窗口。

删除窗口

要删除窗口

1. 在**文件**菜单上，单击**删除窗口**。此时会出现**要删除的窗口**对话框，列出当前打开的所有窗口名。

提示 要快速删除某个窗口，请在 “应用程序浏览器” 中，右击窗口名，然后单击**删除**。您也可以右击窗口的任意空白区，然后单击**删除窗口**。

2. 单击要删除的窗口名旁边的复选框。
3. 单击**确定**以关闭对话框并删除选定的窗口。

备注 除非已经备份，否则不能恢复已删除的窗口。程序会提示您确认删除选定的每个窗口名。

复制窗口

要复制一个现有的窗口，则必须打开所需复制的窗口。

要复制窗口

1. 在**文件**菜单上，单击**窗口另存为**。此时会出现**要在新名称下保存的窗口**对话框，列出当前打开的所有窗口名。

提示 要快速复制窗口，请在 "应用程序浏览器" 中右击该窗口名，然后单击**另存为**。您也可以右击窗口的任意空白区，然后单击**窗口另存为**。

---

2. 单击要复制的窗口名旁边的复选框 （只能选择一个窗口名）。此时会出现*窗口名*另存为对话框。



3. 在**新名称**框中，为新窗口输入一个有效的名称。
4. 单击**确定**以关闭对话框并创建复制窗口。

导出窗口

需要创建或维护一个库应用程序或需要在另一个应用程序中快速创建远程标记名引用时，导出窗口将非常有用。要将窗口从一个 InTouch 应用程序移到另一个应用程序中，您必须使用**文件**菜单上的**导出窗口**命令。

如需有关远程标记名引用的详细信息，请参阅第 6 章 "标记名字典"。

注意！ 如果试图使用任何其它复制方法来复制 InTouch 窗口文件，如使用 "文件管理器" 或 Windows "资源管理器" 的复制命令，则可能会损坏应用程序的 "标记名字典" ！

要导出窗口

1. 关闭当前应用程序中的所有窗口。

2. 在**文件**菜单上，单击**导出窗口**。此时会出现**导出到目录**对话框。



3. 找到并选择要将窗口导出到其中的应用程序目录 （文件夹），然后单击**确定**。

4. The 此时会出现**要导出的窗口**对话框。



5. 选择要导出的窗口。

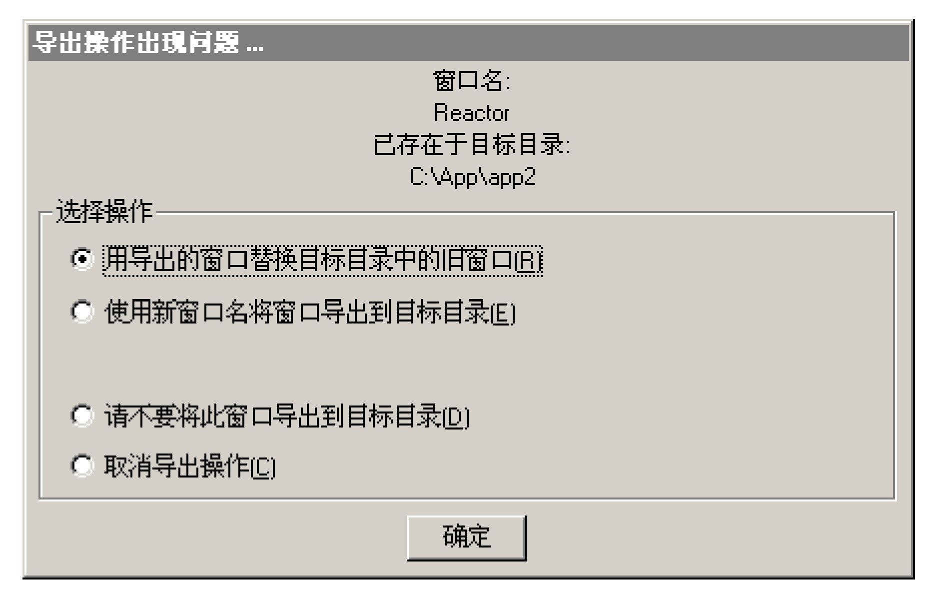
6. 单击**确定**。此时导出操作将开始进行。

备注 导出窗口时，与该窗口关联的所有对象和动画链接将随窗口一起导出。不过，与窗口中的对象关联的标记名会被转换为 “占位符”标记名。通过使用占位符标记名，可以避免在目标应用程序的 “标记名字典”不包含相同标记名时可能出现的任何问题。

如需有关转换占位符标记名的详细信息，请参阅第 6 章 “标记名字典”。

执行导出操作时发生问题

如果在导出窗口时系统遇到问题，则会出现**导出操作问题**对话框。



在**选择操作**组中，选择要采取的措施，然后单击**确定**。

导入窗口

通过将窗口从一个 InTouch 应用程序导入当前应用程序，可以节省您大量的开发时间。它同时还给您提供了一种简便、快捷的方法，供您创建远程标记名引用。它可供您重复使用先前创建的窗口、对象和窗口脚本。要将窗口从一个 InTouch 应用程序移到另一个应用程序中，您必须使用**文件**菜单上的**导入**命令。

如需有关远程标记名引用的详细信息，请参阅第 6 章 “标记名字典”。

备注 如果试图使用任何其它方法移动 InTouch 窗口文件，如使用 “文件管理器”或 Windows “资源管理器”的移动命令，则可能会损坏应用程序的 “标记名字典”！

要导入窗口或 QuickScript

1. 关闭当前应用程序中的所有窗口。

2. 在**文件**菜单上，单击**导入**。此时会出现**浏览文件夹**对话框。




3. 找到并选择包含要从中导入窗口的应用程序目录 （文件夹），然后单击**确定**。

4. 此时会出现 “导入选项”对话框。



5. 选择要导入的项目，然后单击**选择**。此时出现一个对话框，供您选择要导入的窗口或 QuickScript。

6. 选定要导入的窗口或 QuickScript 之后，单击**导入**。系统会自动开始将选定的项目导入当前应用程序。

备注 要导入一个窗口脚本，您必须导入整个窗口。导入窗口时，与该窗口关联的所有对象和链接将随窗口一起导入。不过，与窗口中的对象关联的标记名 （以及导入的脚本中使用的标记名）将转换为 “占位符”标记名。

导入的脚本或窗口中的标记名转换为占位符标记名时，每个标记名的开头会添加三个索引字符。例如，在导入一个离散型标记名时，该标记名的开头会加上三个字符 **?d:** 做前缀。导入长度为 30、31 或 32 个字符的标记名时，每个标记名的开头仍会添加这三个索引字符。不过，增加这三个字符不会截短现有标记名的长度。例如，仅对于占位符标记名，32 个字符标记名会增加到 35 个字符。这三个额外的空间仅分配给占位符标记名。标准标记名不支持这种标记名长度的增加。

如需有关转换占位符标记名的详细信息，请参阅第 6 章 “标记名字典”。

备注 从包含 SuperTag 的应用程序导入窗口时，只有所导入的窗口中实际使用的 SuperTag 实例会导入新应用程序。整个 SuperTag 模板结构不会被导入。例如，如果应用程序定义了数百个 SuperTag 成员标记名，其中只有 50 个用在导入窗口中，则只有这 50 个标记名会被导入。

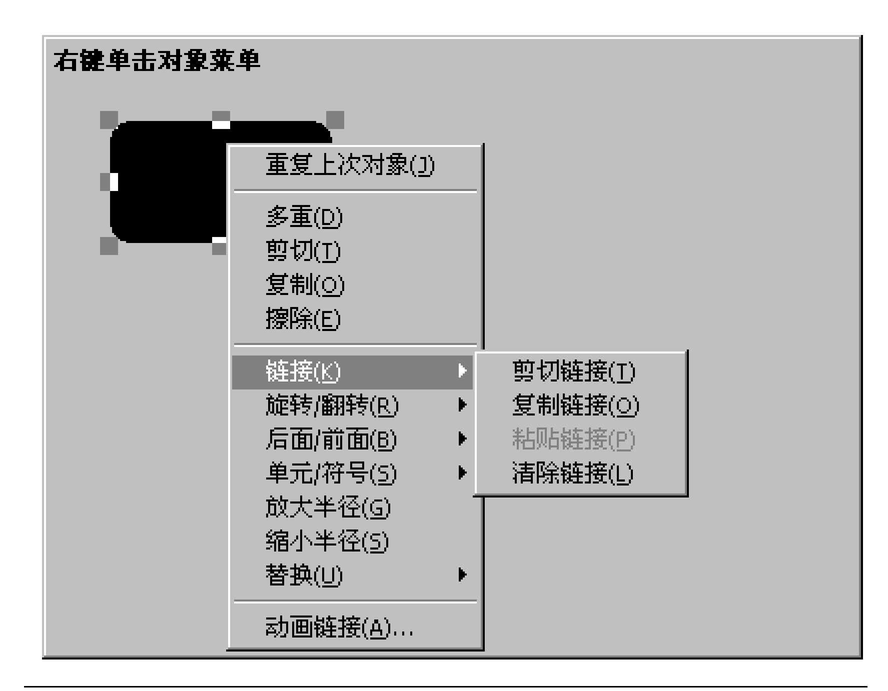
使用图形对象

在应用程序中创建新窗口之后，您就可以使用图形对象来丰富其内容了。 WindowMaker 提供了许多工具来编辑和排列您绘制和粘贴到应用程序在的各种图形对象。本节介绍对图形对象执行各种编辑功能时要采用的操作程序。

对象	描述
绘图对象工具栏	包含用于在窗口在创建图形的绘图工具。
视图工具栏	包含 “标尺”工具，可用于显示标尺来帮助对齐窗口中的图形对象。此工具栏还包含用于在窗口在隐藏或显示 “应用程序浏览器”、工具栏或可见网格的工具。它还包含切换全屏幕模式的工具。
通用工具栏和格式工具栏	包含各种工具，可供您单击用以将**编辑**菜单和**文本**菜单上的许多命令应用于图形对象。
排列工具栏	包含各种工具，供您单击用以快速应用**排列**菜单上的对齐命令。
向导 /ActiveX 工具栏	通过添加重复使用的向导或 ActiveX 控件，您还可以自定义此工具栏。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

提示 如果右击某个对象，则会出现一个菜单，显示可应用于所选对象的有效命令或动作。例如：



备注 右击菜单上的命令会变化。它们取决于您所选的对象类型。

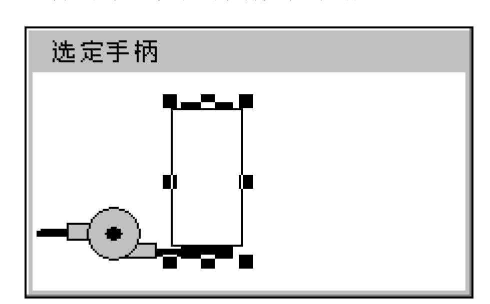
选择对象和调整对象大小

如果在绘制对象后单击它，则它的周围会出现几个小方框。这些方框被称为 “手柄”。您可以使用这些手柄来调整对象大小和 （或）改变它的形状。

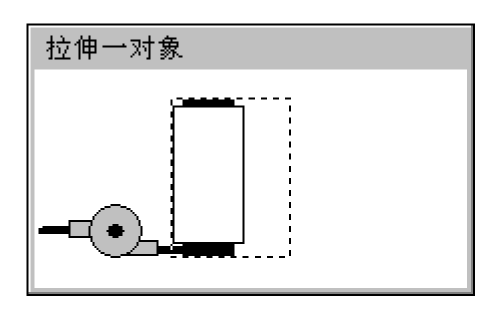
“选定”这个词是 WindowMaker 图形编辑中的一个重要概念。如果对象的周围出现 “手柄”，则表示该对象已被 “选定”。直接单击对象就可选定它。单击窗口的空白区可取消选择该窗口中当前选定的任何对象。一般说来，执行的任何命令均可应用于所有的选定对象，只要该命令对于此对象而言是有效的。

要改变对象的大小

1. 选择对象，然后将箭头光标指针置于 “手柄”中央。

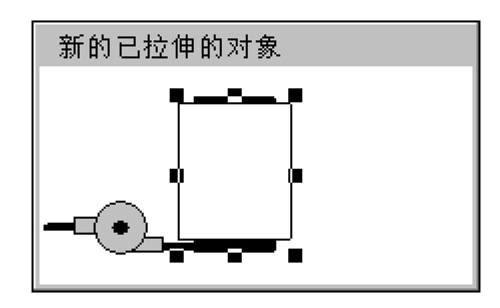


2. 按住鼠标左键不放，同时拖动手柄以缩放对象：



取决于选定的手柄，您可以朝任意方向缩放对象。如果使用四个角手柄之一来改变对象大小，则对象会同时在垂直和水平方向上改变大小。

3. 释放鼠标按钮。此时该对象即被调整至新尺寸：



如果尺寸不正确，请在**编辑**菜单上，单击**撤消**，或单击**通用工具栏**上的 “撤消”工具，然后重试。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

如需有关撤消和恢复编辑的详细信息，请参阅 “撤消对象编辑”。

选择窗口中的所有对象

要选择活动窗口中的所有对象，请在**编辑**菜单上，单击**全选**，或按 F2 键 ；或者，右击打开窗口的空白区，然后单击**全选**。

选择多个对象

要选择多个对象，请在选择第一个对象后，按住 SHIFT 键，并单击要选择的其它对象。要从一组选定的对象中取消选择某个特定的对象，请按住 SHIFT 键，然后单击要取消选择的对象。

选择对象组

把光标移到窗口的空白区。按下鼠标按钮并拖动鼠标。此时会出现一个带小手光标的虚线选择矩形框。拖动鼠标直至完全包围要选定的所有对象。释放鼠标按钮。此时完全在该矩形框内的所有对象均会被选定。

取消选择对象组

如果在按住 **Shift** 键的同时绘制一个选择矩形框，则会取消选择包围的所有对象。此项技巧也可以用来开始在另一个对象上绘制选择矩形框。

如果在按住 **Shift** 键的同时单击鼠标左键，则在移动光标时不能拖动光标下的对象。相反，此时会绘制一个选择矩形框。

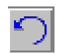
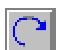
撤消对象编辑

WindowMaker 可以跟踪所作的编辑和格式更改。您可以配置 WindowMaker，使之支持最多 25 级撤消 / 恢复操作。您也可以通过将级数设置为零 (0) 来停用撤消 / 恢复功能。缺省条件下， WindowMaker 会设置为支持 10 级，其中每级代表一个动作。

备注 创建新窗口或打开现有的窗口时，“撤消”和 “恢复”堆栈是空的。保存窗口时，这两个堆栈也会将被清空。

如需有关配置撤消 / 重复级数的详细信息，请参阅 “自定义开发环境”。

撤消和**恢复**命令均位于**编辑**菜单。这些命令也可以通过右击对象进行访问。这些命令会动态改变，以反映可应用的最后一个动作。例如，如果您移动了某个对象，随后决定恢复它在窗口中的初始位置，则可以右击窗口的空白区，单击**撤消移动**，或者在**编辑**菜单上，单击**撤消移动**。

  “撤消”和 “恢复”工具位于**通用工具栏**中。

您可以使用撤消和恢复操作来反转应用于对象的动作或命令。您可以撤消和恢复以下动作或命令：

命令	动作
基本	创建、选择、取消选择、移动和调整对象线尺寸、填充、文本和窗口颜色、窗口移动及窗口大小
编辑	复制、剪切、拷贝、粘贴、删除粘贴位图和调整位图 - 初始大小、选择全部、链接剪切、链接复制、链接粘贴、链接清除、扩大半径、缩小半径、改变对象形状、添加点、删除点
排列	置后、置前、对齐 （所有命令）、水平间距、垂直间距、顺时针旋转、逆时针旋转、水平翻转、垂直翻转、制作符号、分解符号、制作单元、分解单元
文本	所有操作 （大小、样式、间距、对齐方式）
线条	所有操作 （宽度和样式）
特殊	动画链接 （双击对象）、替换字符串

复制对象

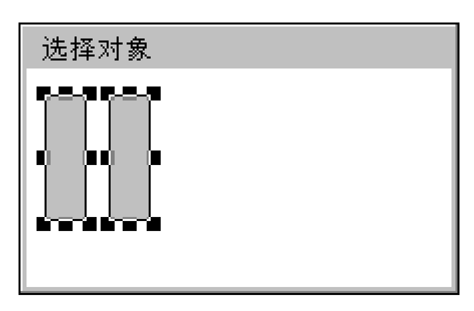
要重绘以前绘制的对象

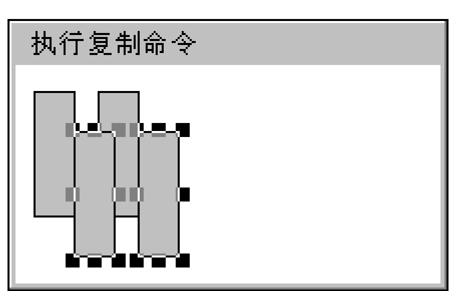
1. 绘制对象。
2. 右击对象，然后单击**重复上次对象**。
3. 单击鼠标左键，然后再次绘制相同的对象。

要复制对象

1. 选择要复制的对象。
2. 在**编辑**菜单上，单击**复制**，或者单击**通用工具栏**上的 “复制选项” 工具。

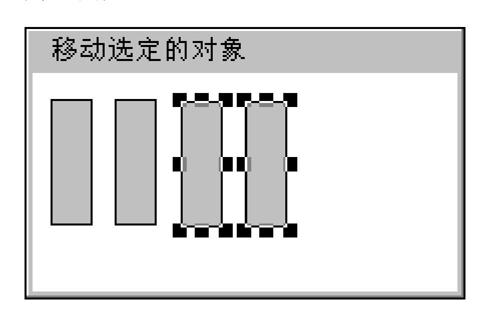
   如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

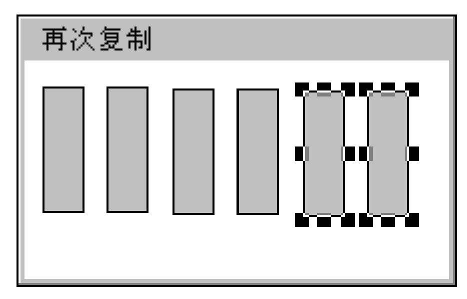




提示 要快速复制对象，请右击该对象然后单击**复制**。

如果在没有取消选择的情况下移动所复制的对象，则再次复制时，第二次复制 （及所有后续复制）会自动偏移第一次复制时移动的距离。例如：





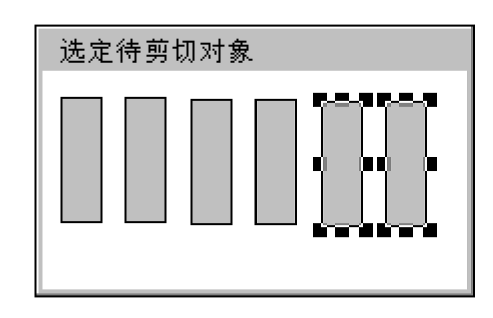
您可以根据需要重复任意多次此操作程序。

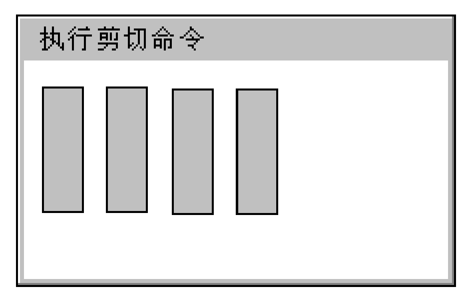
将对象剪切到 Windows 剪贴板

要剪切对象

1. 选择要剪切的对象。

2. 在**编辑**菜单上，单击**剪切**，或者单击**通用工具栏**上的“剪切到剪贴板”工具。





提示 要快速剪切对象，请右击该对象，然后单击**剪切**。

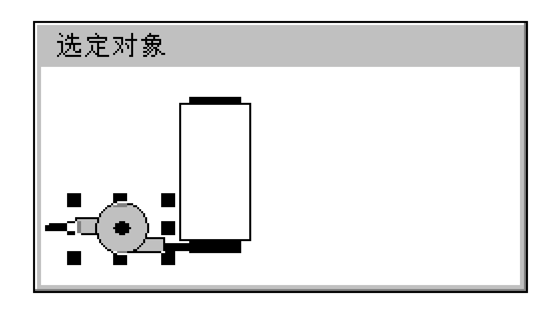
备注 剪切对象时，该对象会从窗口被移除并复制到 Windows “剪贴板”上。该对象的属性和动画链接也会随它一同复制。

将对象复制到 **Windows** 剪贴板

要复制对象

1. 选择要复制的对象。
2. 在**编辑**菜单上，单击**复制**，或者单击**通用工具栏**上的“复制到剪贴板”工具。

   如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。



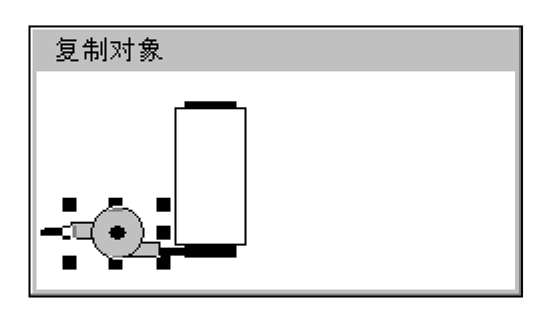
提示 要快速复制对象，请右击该对象，然后单击**复制**。

备注 复制对象时，它不会从窗口中被删除。它是被复制到 Windows “剪贴板”上。该对象的属性和动画链接也会随它一同复制。

从 Windows 剪贴板粘贴对象

要从 Windows 剪贴板粘贴对象

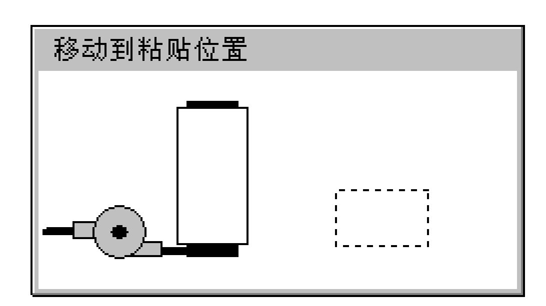
1. 复制或剪切对象：



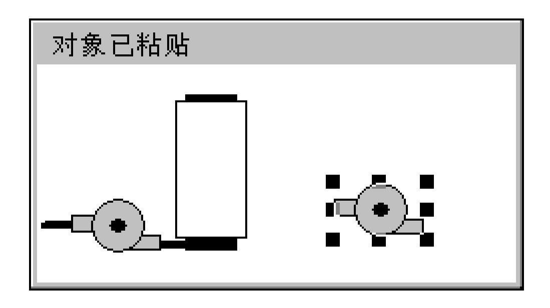
2. 在**编辑**菜单上，单击**粘贴**，或者单击**通用工具栏**上的 “从剪贴板粘贴”工具。

提示 要快速粘贴所复制的对象，请右击该对象，然后单击**粘贴**。

3. 光标将变成一个弯头符号。
4. 按住鼠标左键，此时会出现一个所复制对象大小的虚线矩形框。将矩形框拖到要将对象对象到其中的窗口中的某个位置。



5. 释放鼠标按钮以粘贴对象：



提示 粘贴的所有对象在粘贴后仍保持选定状态，您可以通过移动来调整它们的位置。

备注 将对象复制或剪切到 Windows “剪贴板”时，该对象的所有属性和动画链接也将随之一起复制。如果随后将该对象粘贴到窗口，它所有的属性仍将保持原样。

剪切和粘贴对象链接

WindowMaker 的链接粘贴缓冲区是一个临时存储区，用于存储从一个对象剪切或复制的链接。（缓冲区只存储最近的剪切或复制动作的链接）。您可以将链接粘贴缓冲区中存储的链接粘贴到任何对象或符号上。如果选择多个对象，则该链接将粘贴到每个不同的对象上。

如果粘贴的链接对于对象没有明显的值，例如文本对象上的线颜色链接，则不会粘贴该链接。

如需有关动画链接的详细信息，请参阅第 7 章 “创建动画链接”。。

要剪切、复制、粘贴和清除链接

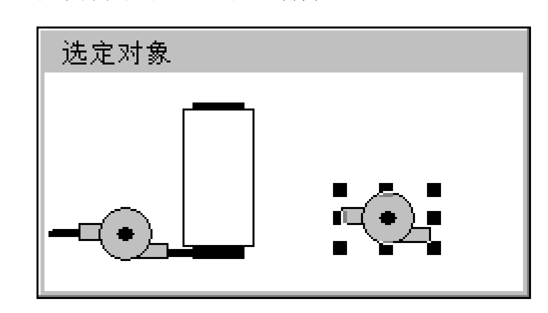
1. 选择要应用链接命令的对象。
2. 在**编辑**菜单上，指向**链接**，然后单击相应的命令。

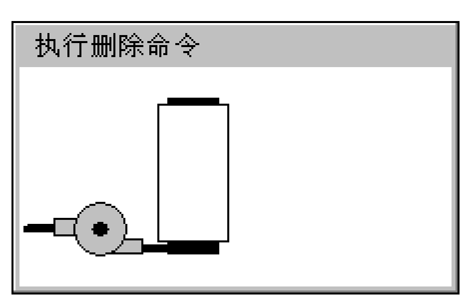
提示 要快速访问链接命令，请右击该对象，指向**链接**，然后单击相应的链接命令。

删除对象

要删除对象

1. 选择要删除的对象。
2. 在**编辑**菜单上，单击**清除**。





提示 要快速删除对象，请右击该对象，然后单击**清除**，或在选定对象后按 DEL 键。

备注 被删除的对象不会复制到 Windows “剪贴板”。

增大或减小圆角对象的半径

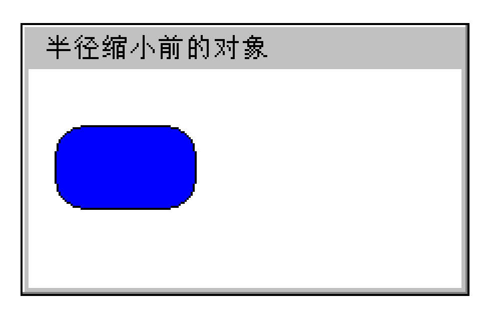
您可以增大和 （或）减小使用 “圆角长方形”工具绘制的任何对象的角半径。

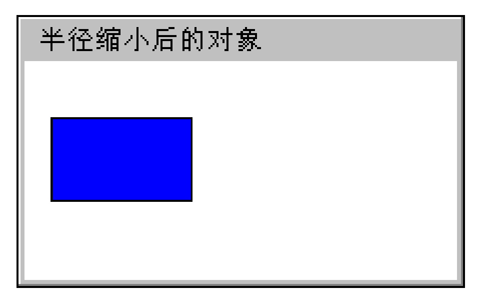
要增大 （或减小）圆角对象的半径

1. 选择对象。
2. 在**编辑**菜单上，单击**放大半径** （或**缩小半径**）。

提示 要快速增大或减小半径，请右击该圆角对象，然后单击相应的命令。

3. 重复此命令，直至半径增加到所需大小。例如：



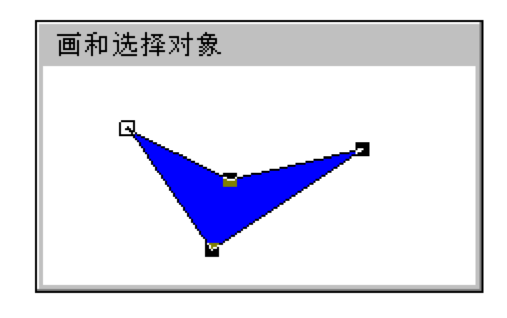


提示 您也可以使用键盘快捷键 Shift+ 加号 （数字小键盘上的 + 号）来增大半径，或 Shift + 减号 （数字小键盘上的 - 号）来减小半径。每次按下这些组合键时，便会执行相应的命令。如果按下该键不放，则会不停执行命令直到对象的半径不能再增减为止。

改变多边线或多边形对象的形状

要调整多边线或多边形的形状

1. 选择多边线或多边形对象。



提示 绘制对象时鼠标单击的每个 “点”会重新显示为 “手柄”。

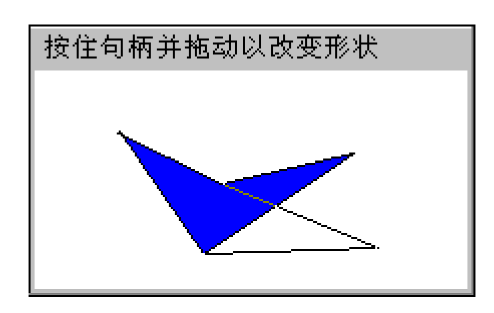
---

2. 在**编辑**菜单上，单击**改变对象形状**，或在**排列工具栏**上单击 “改变对象形状”工具。

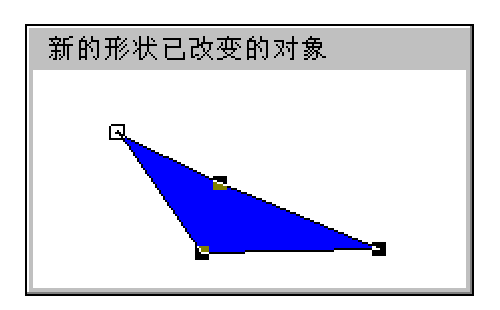
提示 要快速调整多边线或多边形的形状，请右击对象，然后单击相应的命令。

---

3. 要调整对象形状，请抓住手柄并将它拖到所需的位置：



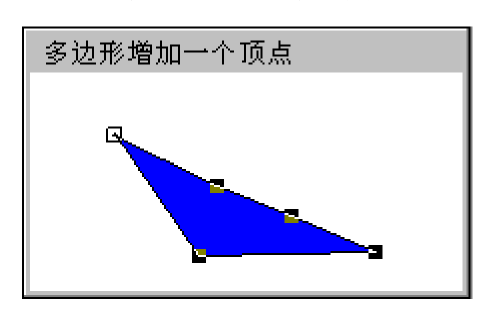
4. 释放鼠标，该对象即被调整为新的形状：



要添加或删除多边线或多边形上的顶点

1. 选择多边线或多边形对象。

2. 在**编辑**菜单上，单击**添加顶点**或**删除顶点**，然后单击希望添加新点到对象的位置，或单击希望删除的点。

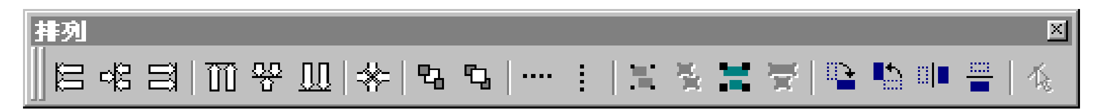


提示 要在多边线或多边形上快速添加（或删除）顶点，请右击对象，然后单击相应的命令。

排列窗口中的对象

WindowMaker 提供了许多用于在窗口中排列对象的工具。本节介绍 WindowMaker 提供的各种排列工具。

提示 通过使用**排列工具栏**中包含的工具，您可以给选定的对象快速应用**排列**菜单上的大多数命令。例如：



如需有关**排列工具栏**的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

对齐对象

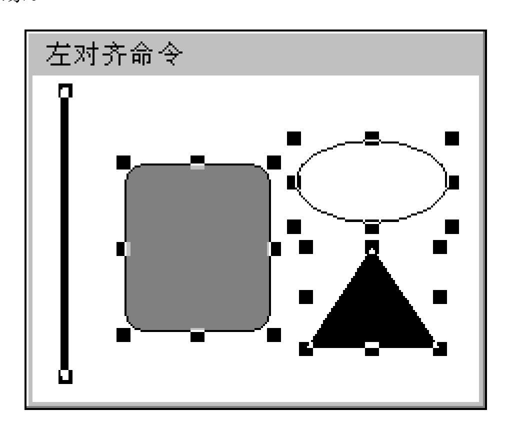
您可以根据对象的左或右边缘、中心、中心点、顶部、中间或底部来对齐它们。

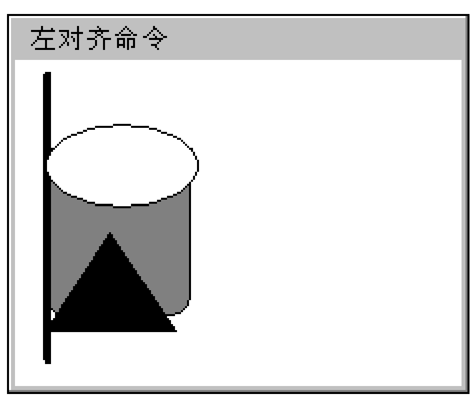
要对齐选定的所有对象

1. 选择对象。
2. 在**排列**菜单上，指向**对齐**，然后单击相应的对齐命令。此时选定的对象将根据您的选择进行对齐。

下面的几个例子显示每个对齐命令的行为方式：

左对齐以组中最左端对象的左边缘为基准，对齐所有选定对象的左边缘。

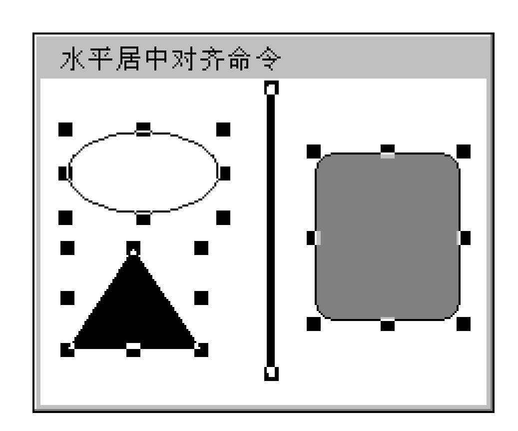


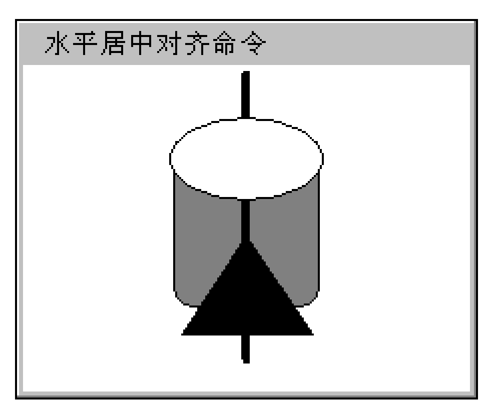


提示 要快速对齐对象，请选择对象，然后单击**排列工具栏**上的相应工具。

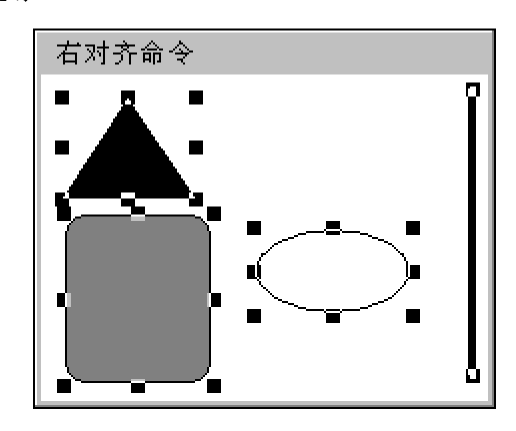
如需有关工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

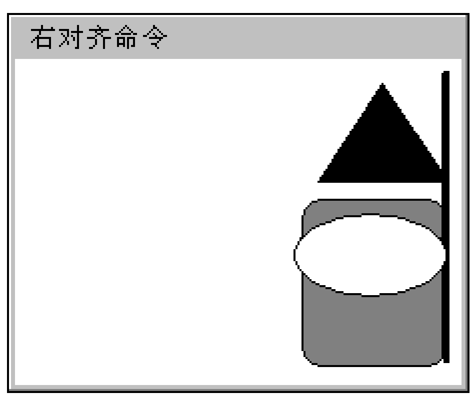
中心对齐以组中的垂直中心线为基准，对齐所有的选定对象：



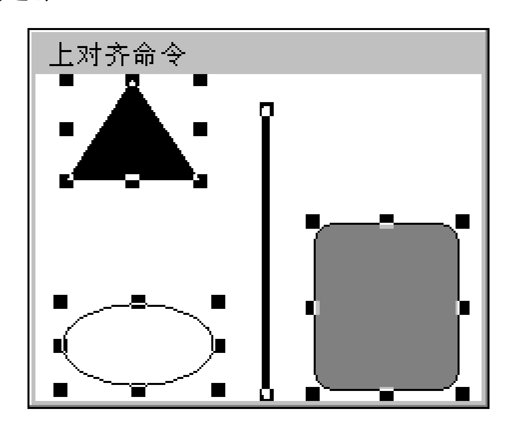


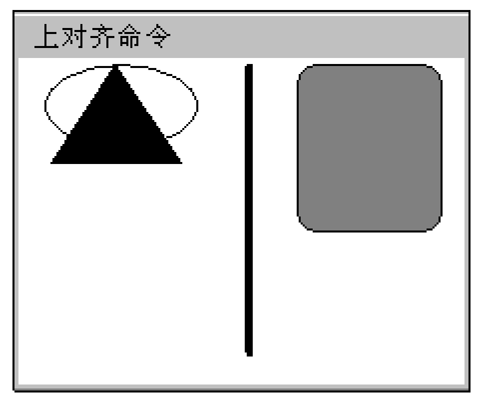
右对齐以组中最右端对象的右边缘为基准，对齐所有选定对象的右边缘：



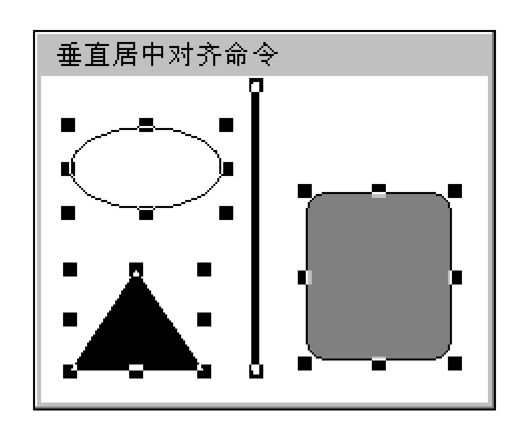


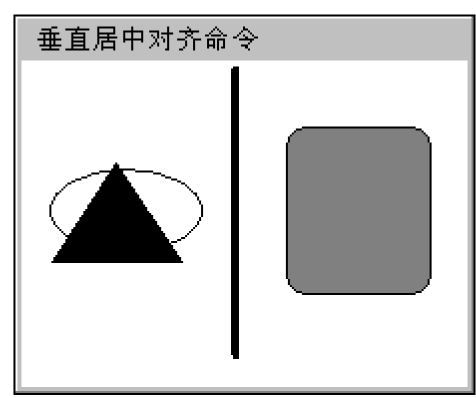
顶部对齐以组中最顶端对象的顶边缘为基准，对齐所有选定对象的顶边缘：



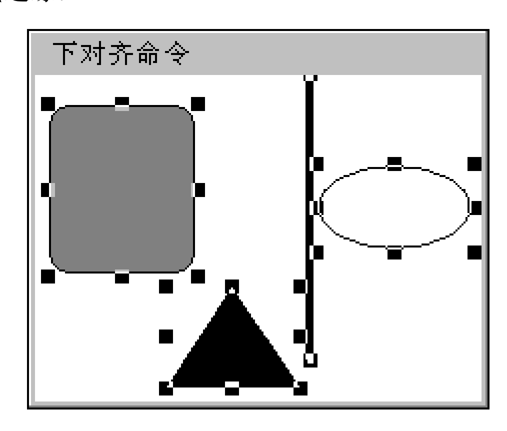


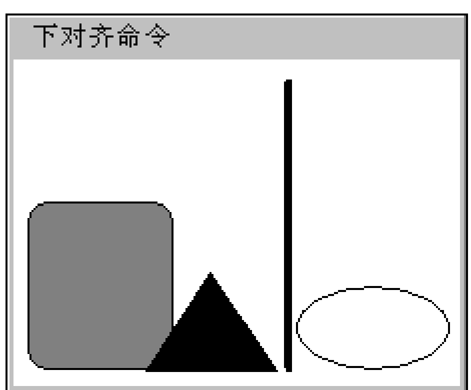
垂直居中对齐以组的中间位置为基准，对齐所有选定对象的中部：



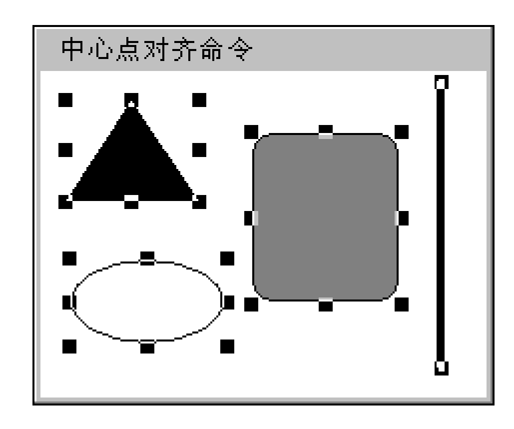


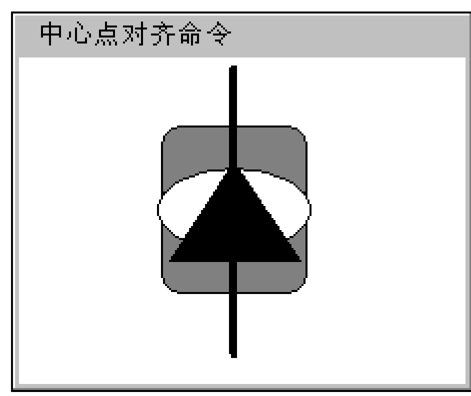
底部对齐以组中最底端对象的底边缘为基准，对齐所有选定对象的底边缘：





中心点对齐以组的中心点为基准，对齐所有选定对象的中心点：





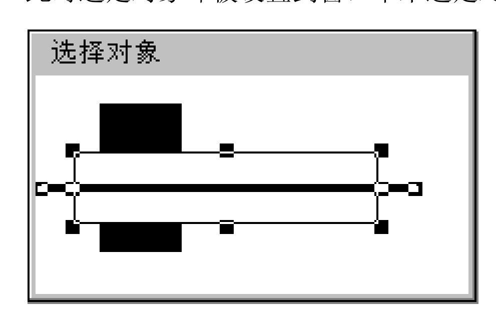
对象分层

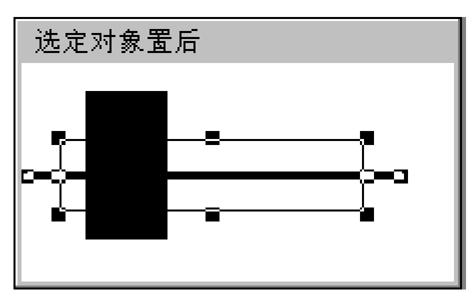
您可以通过将对象置前或置后而将窗口中的对象分层。

要将一个对象置于另一对象之后

1. 选择对象。

2. 在**排列**菜单上，单击**置后**，或单击**排列工具栏**上的“置后”工具。此时选定对象即被设置到窗口中未选定对象的后面：

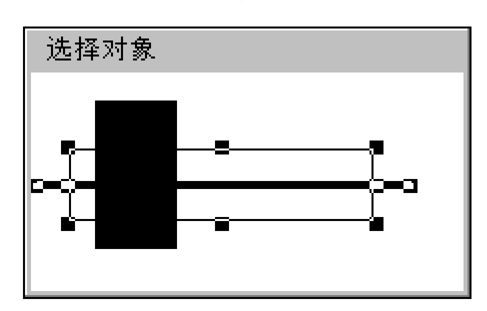


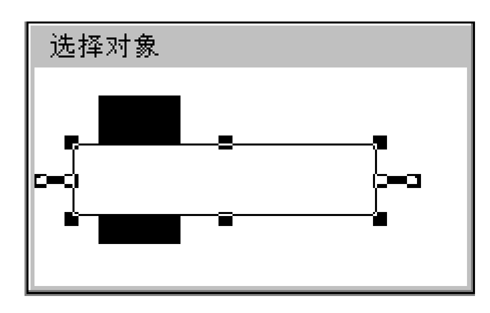


提示 要快速将对象置后，请右击对象，指向**前 / 后**，然后单击**置后**。

要将一个对象置于另一对象之前

1. 选择对象。
2. 在**排列**菜单上，单击**置前**，或单击**排列工具栏**上的“置前”工具。此时选定的对象即被设置到窗口中未选定对象的前面：





请注意最后的对象被置于最前端。

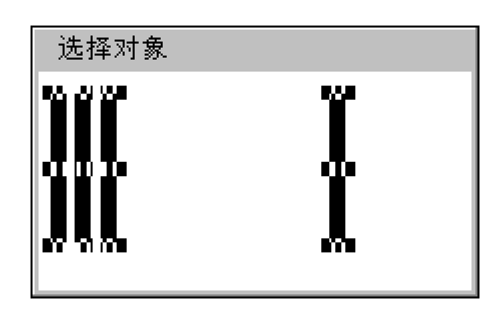
提示 要快速将对象置前，请右击对象，指向**前 / 后**，然后单击**置前**。

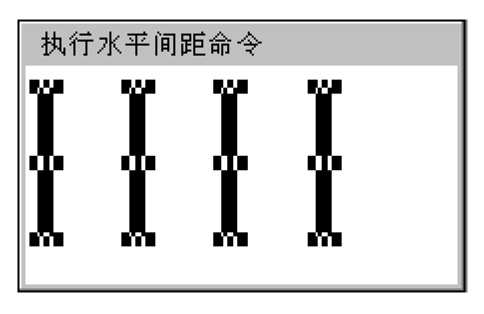
控制水平和垂直间距

您可以在最左端与最右端的选定对象之间沿水平方向均匀分布对象。您也可以控制在最顶部和最底部的对象之间的垂直间距。

要水平或垂直均匀分布对象

1. 选择对象。
2. 在**排列**菜单上，单击**水平间距**（或**垂直间距**），或单击**排列工具栏**上的相应工具。此时选定的对象即被设置为在相隔最远的两个对象之间均匀分布。例如：





如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

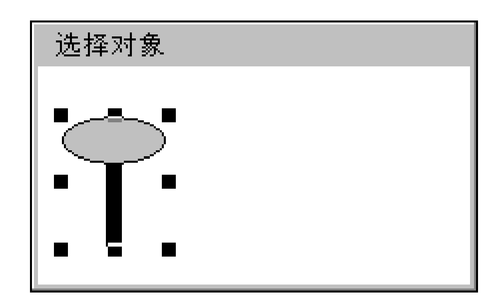
旋转对象

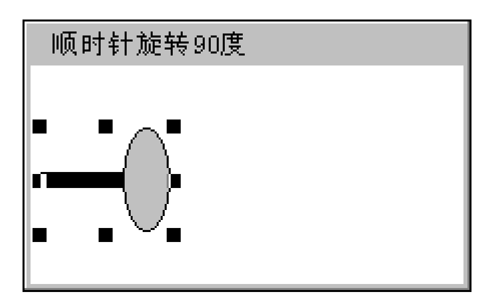
在 WindowMaker 中，您可以旋转大多数对象，包括位图、JPEG、PCX、TGA 图像和文本对象。对象能以 90 度的增量顺时针或逆时针旋转 360 度（即 90 度、 180 度、 270 度和 360 度）。对象附带的任何链接也将随之一起旋转。您不能旋转单元。不过，您可以旋转符号。

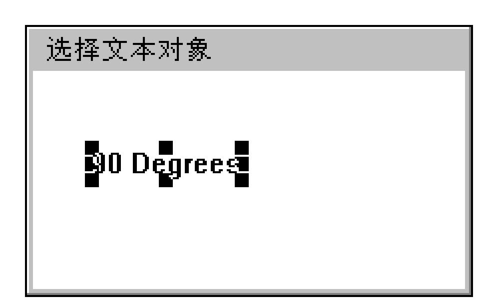
备注 在 WindowMaker 中旋转对象与在运行时动态旋转对象无关。在 WindowViewe 中，对象通过链接**方向**动画链接来旋转。文本对象不能在 WindowViewer 中旋转。不过，位图或图像可以通过指定**方向**动画链接进行旋转。

要将选定对象旋转 90 度

1. 选择对象。
2. 在**排列**菜单上，单击**顺时针旋转** （或**逆时针旋转**）。此时选定的对象将按选定的方向旋转 90 度。









要使对象旋转 180 度，请重复此操作程序。要将对象旋转 270 度，请执行此操作程序两次，以此类推。

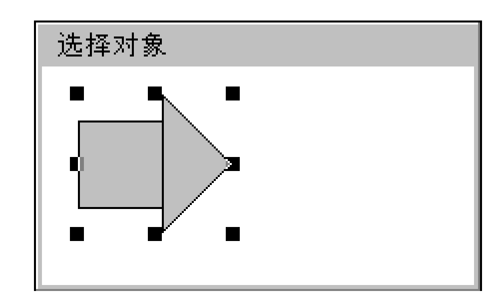
提示 要快速旋转对象，请右击对象，指向**旋转 / 翻转**，然后单击相应的命令。

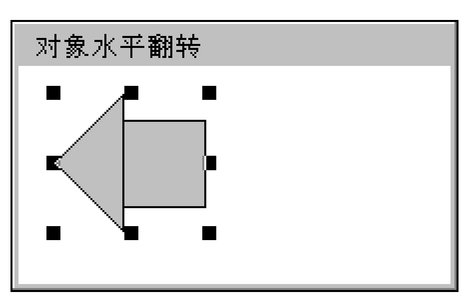
对象镜像

您可以水平或垂直翻转大多数 WindowMaker 对象，包括位图、 JPEG、PCX 和 TGA 图像 （文本只能旋转而不能翻转）。在翻转对象时，您可以将它变换成水平或垂直的镜像。对象附带的任何链接也将随之一起翻转。

要翻转选定的对象

1. 选择对象。
2. 在**排列**菜单上，单击**水平翻转** （或**垂直翻转**），或选择**排列工具栏**上的相应工具。此时选定对象将翻转。例如：





提示 要快速旋翻对象，请右击对象，指向**旋转 / 翻转**，然后单击相应的命令。

如需有关工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

创建单元和符号

您可以将多个对象合并成两种不同类型的单元：单元和符号。多个单元可以合并成一个单元。单元是在不同图形元素之间保持固定空间关系的对象。单元内的单个组件 （另一个单元除外）可以设置动画效果。单元不能调整大小，也不能附加动画链接。不过，符号可以附带动画链接，而单元又可以包括符号。与单元在的符号或对象关联的所有动画链接均不得改变。单元在对象的属性如文本、字体、行宽、半径及相对位置不能调整大小，直到单元被分解成单个组件。

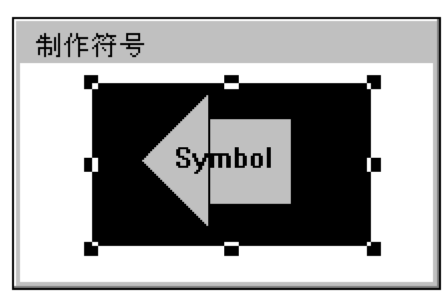
提示 双击单元时会出现**替换标记名**对话框 （不是对象和符号的动画链接选择对话框）。

如需有关替换标记名的详细信息，请参阅第 6 章 “标记名字典”。

备注 合并单元时，每个单元均会被保留，所以在分解合并的单元时，它们会恢复为原始的单元。

符号可由多个符号和 （或）多个简单对象组成，如下图所示：





如果某个选定对象附带动画链接，则该链接会附加到新符号上。（如果链接粘贴缓冲区包含链接，程序会询问是否要将该链接粘贴到新符号上）。

备注 如果多个选定的对象包含链接，则不能制作符号。如果将两个符号合并成一个新符号，则原始符号结构会丢失。因此，如果分解新符号，则它会分解成每个原始符号的单独组件。此时两个原始符号便会丢失。

如需有关动画链接的详细信息，请参阅第 7 章 “创建动画链接”。。

要创建符号或单元

1. 选择要包含到单元或符号中的对象。
2. 在**排列**菜单上，单击**制作单元** （或**制作符号**），或单击**排列工具栏**上适当的工具。

提示 要快速创建单元或符号，请选择所有的对象。右击选定对象的之一，指向**单元 / 符号**，然后单击相应的命令。

要分解符号或单元

1. Select the symbol or cell...
2. 在**排列**菜单上，单击**分解单元** （或**分解符号**），或单击**排列**工具栏上适当的工具。

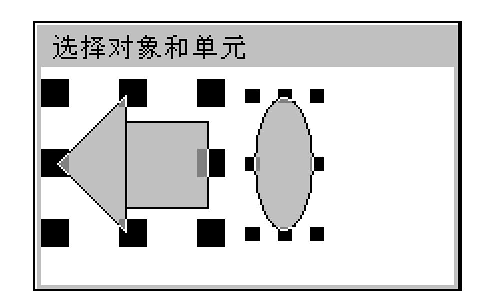
提示 如果给符号定义了链接，则这些链接会自动保存到链接粘贴缓冲区。

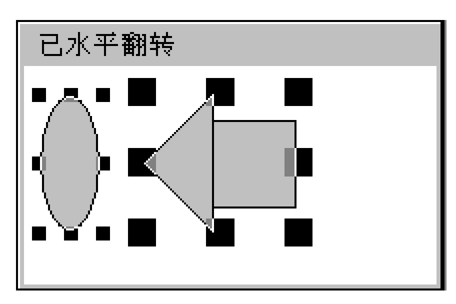
要快速分解单元或符号，请右击单元或符号，指向**单元 / 符号**，然后单击适当的命令。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

翻转单元

翻转单元时，它们没有被映射。映射的只有对象组中的单元位置。例如：





比较单元翻转前后的位置 （左边）及其朝向。它的位置翻转了，但内容没有翻转。椭圆对象发生了映射。这同样适用于单元在垂直方向上的翻转。

将对象对齐网格

在窗口中排列对象时，打开网格可以让图形按网格左上角的像素间隔进行对齐。如果选择多个对象，则会按组中第一个选定对象的左上角来对齐网格。

单击**视图**工具栏上的 “对齐网格”工具，以打开或关闭对齐网格功能，或在**排列**菜单上，单击**对齐网格**。

提示 缺省条件下，网格设置为 10 个像素大小，并在初次启动 WindowMaker 时是可见的。您可以通过 **WindowMaker 属性**对话框来配置网格的像素间隔。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

要配置网格

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowMaker**，或在 “应用程序浏览器” 中的**配置**下，双击 **WindowMaker**。此时会出现 **WindowMaker 属性**对话框。
2. 在**间距**框中，输入对齐网格坐标间隔的像素数。
3. 如果要在打开 WindowMaker 的 “对齐网格” 功能时在窗口中显示网格，请选择**显示网格**。

提示 如果不选择**显示网格**，则打开对齐网格功能时将不能在窗口中看见网格。

如需有关配置网格的详细信息，请参阅 “自定义开发环境”。

使用图像与位图

在其它 Windows 程序中创建的所有图形对象 （如图形、屏幕截图、AutoCad 图形、 JPEG、 PCX 和 TGA 文件类型等）必须粘贴到 WindowMaker 的位图容器中。

WindowMaker 将位图视为单个对象。因此，您不能给位图中的单个元素设置动画，也不能将位图放入符号中。不过，您可以将位图包含到单元中。

在 WindowMaker 中，您可以旋转位图、JPEG、PCX 和 TGA 图象。它们能以 90 度的增量顺时针或逆时针旋转 360 度 （即 90 度、 180 度、 270 度和 360 度）。位图附带的任何链接将随之一起旋转。

备注 在 WindowMaker 中旋转位图与在运行时动态旋转位图无关。在 WindowViewer 中，位图或图像通过链接**方向**动画链接进行旋转。

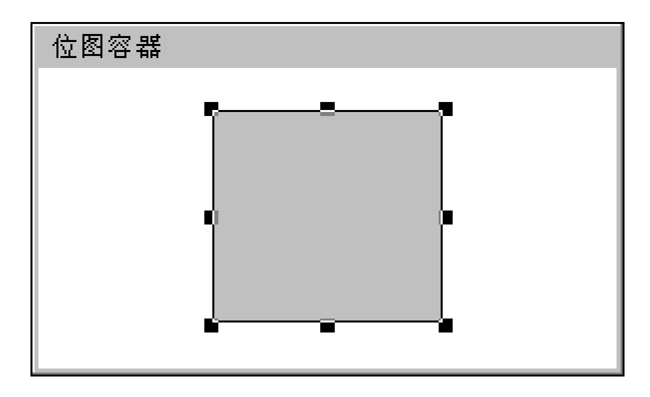
您也可以使用透明颜色定义位图，这样您就可以让它浮在其它对象上。使用透明颜色定义位图时，使用透明颜色的所有位置将显示窗口背景色或位图后面的任何对象。（每个位图只能使用一种透明颜色）。

如需有关透明位图与图像的详细信息，请参阅 “创建透明位图”。

要导入位图或 JPEG、 PCX 或 TGA 文件类型

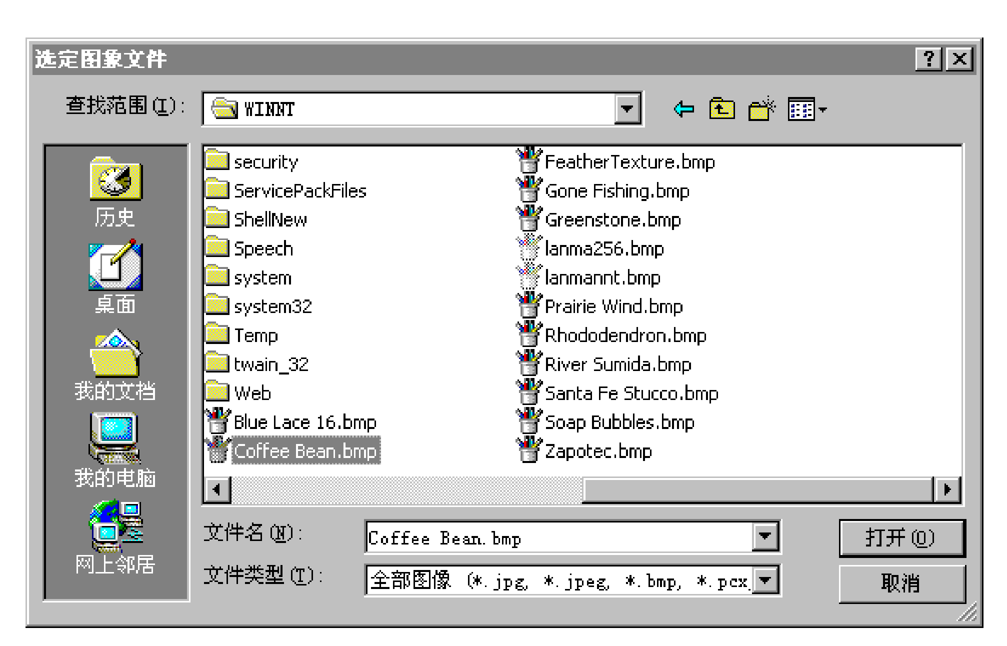
1. 单击 “位图” 工具 （光标将变成十字形），然后在窗口中绘制一个位图容器 （任意大小）。

2. 选择位图容器：

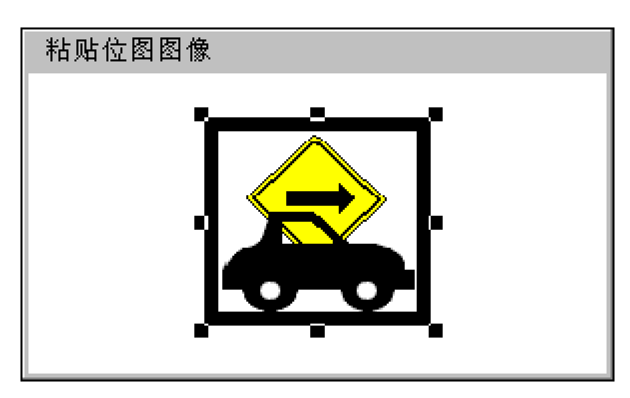


3. 在**编辑**菜单上，单击**导入图像**。此时会出现 Windows **选择图像文件**对话框。

提示 要快速粘贴图像，请右击位图容器，然后单击**导入图像**。



4. 找到并选择要作为位图导入的 .BMP、 PCX.TGA 或 .JPG 文件，然后单击**打开**或双击图像文件名。此时该图像将粘贴到位图容器中。



5. 要使用原始大小的位图，请选择它，然后在**编辑**菜单上，单击**位图 - 原始大小**。此时该位图按照它原来的尺寸重绘。

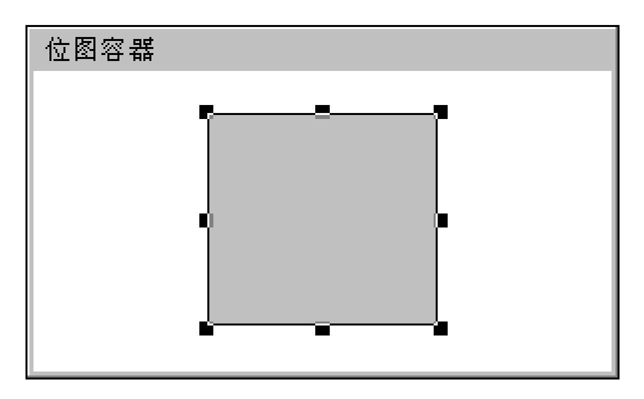
提示 要快速调整位图大小，请右击该位图，然后单击**位图 - 原始大小**。



从 Windows 剪贴板粘贴位图

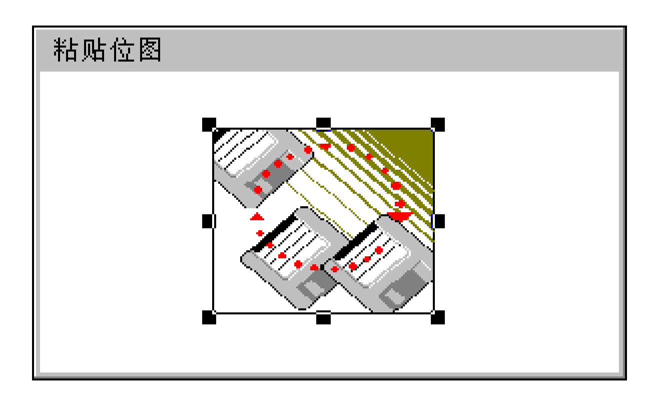
要将位图从 Windows 剪贴板粘贴到窗口

1. 将图形复制到 Windows “剪贴板”。例如，显示图形，在按 ALT 键的同时按下 PRINT SCRN 键，将它复制到 Windows “剪贴板”。
2. 单击 “位图”工具 （光标将变成十字形），然后在窗口中绘制一个位图容器 （任意大小）。
3. 选择位图容器。

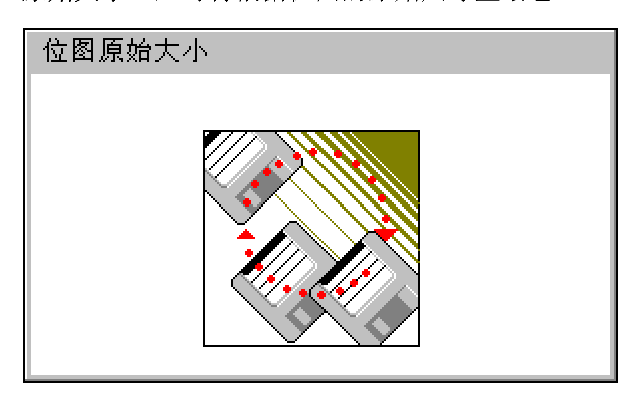


4. 在**编辑**菜单上，单击**粘贴位图**。此时该位图会从 Windows “剪贴板”粘贴到位图容器中：

提示 要快速粘贴位图，请右击该位图容器，然后单击**粘贴位图**。



5. 要使用原始大小的位图，请选择它，然后在**编辑**菜单上，单击**位图 - 原始大小**。此时将根据位图的原始尺寸重绘它。



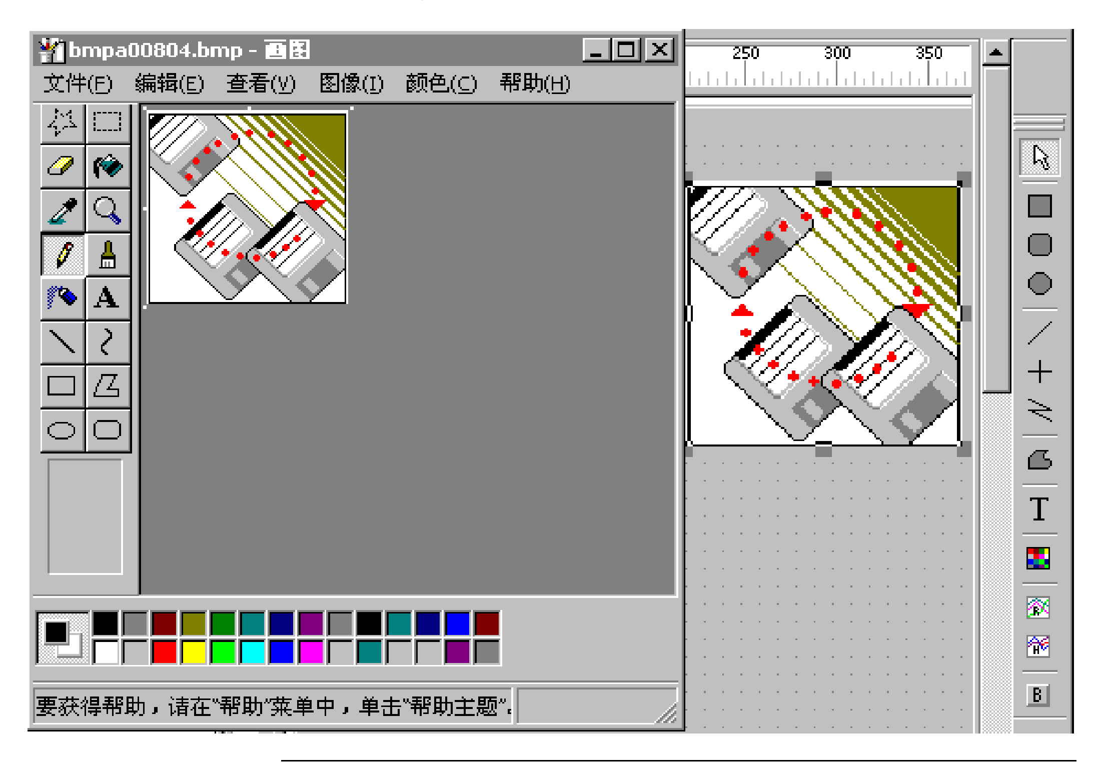
提示 要快速调整位图大小，请右击该位图，然后单击**位图 - 原始大小**。

编辑位图

您可以直接从 WindowMaker 中编辑位图。在 WindowMaker 中编辑位图时，将会启动与 .bmp 扩展名关联的应用程序。例如，如果位图是使用 MSPaint 创建的，则在 WindowMaker 中编辑位图时， MSPaint 会自动启动，具体如下所述：

要在 WindowMaker 中使用 Windows Paint 编辑位图

1. 选择位图。在**编辑**菜单上，单击**编辑位图**。此时 Microsoft Paint 会打开并显示该位图：



提示 要快速编辑位图，请右击该位图，然后单击**编辑位图**。

2. 在 MS Paint 中编辑位图。
3. 退出 MS Paint 以返回 WindowMaker。

备注 在使用原来的应用程序编辑位图时，您将不能使用 WindowMaker。要返回 WindowMaker，请退出原始应用程序。

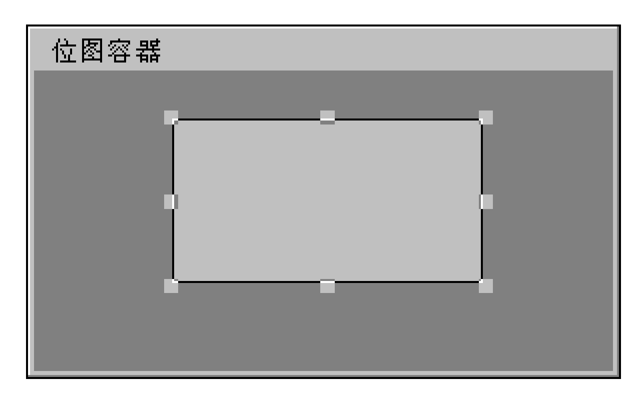
创建透明位图

您可以用透明颜色定义位图或图像，这样就可以让它漂浮在其它对象上。使用透明颜色定义位图或图像时，使用透明颜色的所有位置将显示窗口的背景色或位图后面的任何对象。

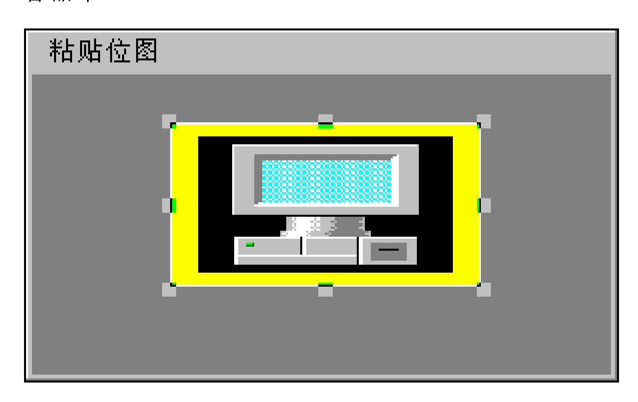
要创建透明位图

1. 单击 “位图” 工具 （光标将变成十字形），然后在窗口中绘制一个位图容器 （任意大小）。

2. 选择位图容器：



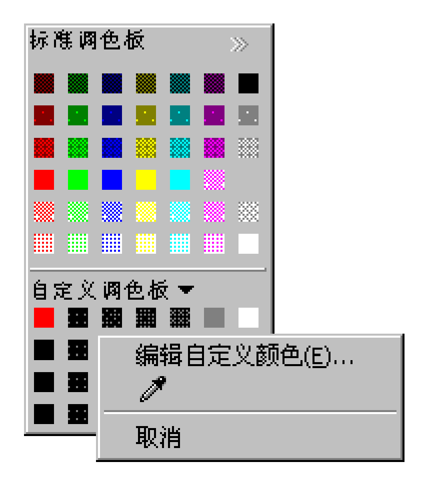
3. 右击位图容器，然后单击**粘贴位图**（如果已将该图形复制到 Windows“剪贴板”）；否则，请单击**导入图象**（找到并选择要打开的 .BMP、.PCX、.TGA 或 .JPG 文件）。此时位图图像将粘贴到位图容器中：



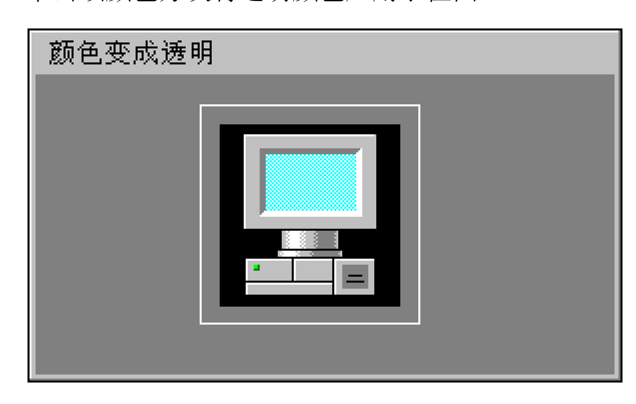
4. 右击该位图，然后单击**位图 - 原始大小**，让位图恢复其原始大小。

5. 选定位图，然后单击**格式**工具栏上的“透明颜色”工具，以打开透明色调色板。

6. 右击调色板底部**自定义调色板**区域中的空白颜色方块。此时会出现**编辑自定义颜色**对话框。



7. 单击 “吸管”工具 （**编辑自定义颜色**对话框将关闭）。
8. 单击位图中要设置为透明的颜色。此时该颜色将复制到透明色调色板中选定的颜色方块上。
9. 单击该颜色方块将透明颜色应用于位图。



在本例中，我们将位图的宽边区域设置为透明。因此，现在可以透过位图显示出窗口的背景色。如果对象位于现在的透明区域之后，则它们也可以透过位图显示出来。

备注 一个位图只能应用一种透明颜色。

使用文本对象

在 WindowMaker 中，您可以改变任意选定文本对象的字体、字形、字体大小、对齐方式和旋转。您也可以按 90 度的增量将文本对象顺时针或逆时针旋转 360 度 （90 度、 180 度、 270 度和 360 度）。

例如：





备注 在 WindowMaker 中旋转文本对象与在运行时动态旋转对象无关。**方向**动画链接不适合文本对象。因此，文本对象不能在 WindowViewer 中旋转。

通过使用**格式工具栏**中包含的工具，您可以给选定的对象快速应用**文本**菜单上的大多数命令。例如：



如需有关 “格式工具栏” 的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

设置文本对象格式

所有的 WindowMaker 文本命令都是在单个或多个文本字符串选项及与数值字段上执行的。如果在执行**文本**菜单上的命令时没有选定文本对象，则该命令会自动应用于**格式工具栏**上相应的文本工具的缺省设置，以及**绘图对象工具栏**上 “文本” 工具的缺省设置。

对于用来输出动态值的文本对象而言，文本对齐方式属性设置特别重要。对齐方式确定了在运行时如何显示不同长度的字段。

例如，如果要在居中或右对齐的文本字符串末尾显示数值，则每当显示的数字位数改变时，整个文本字符串 （包括值）将会再次居中或右对齐。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

显示数值

文本对象也可用于显示静态或动态数值。通过给文本对象附加**触动链接用户输入 - 模拟**或**值显示 - 模拟**动画链接，您可以显示模拟 （整型或实型）标记名的值。

要确定模拟量的显示格式，可使用下面的四个字符：

字符	描述
0	零
#	数字或磅符号
,	逗号
.	句点或小数点

下面举例说明模拟量的字段格式设置：

字符	描述
#	显示任意整数。例如： **1234** 会显示成 **1234** （只需一个 # 号）
0.0	强制使用一个前导零和一个小数位。例如： **.1** 会显示成 **0.1** **77.1** 会显示成 **77.1**
00000	根据需要强制使用前导零。例如： **123** 会显示成 **00123** **1234** 会显示成 **01234** **12345** 会显示成 **12345**
#,##0.0	如有需要，请插入逗号和前导零，以及一个小数位。例如： **1234.56** 会显示成 **1,234.6** **123.4** 会显示成 **123.4**
0,000.0	强制使用逗号、前导零和一个小数位。例如： **12.3** 会显示成 **0,012.3**

备注 如果在格式中使用零，则它后面必须跟零。小数点右边所有的位必须始终为零。例如， 000.00 正确，而 #0#0.0# 则不正确。

提示 所有常规文本格式均适用于数值。这些包括字体、大小、颜色、对齐方式及粗细。

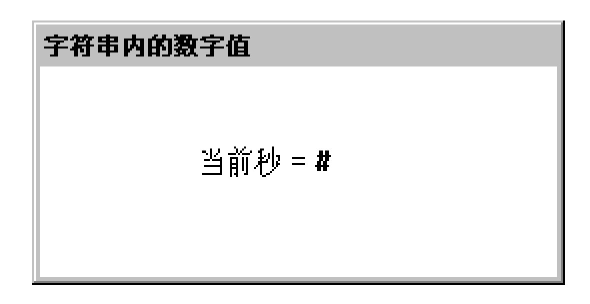
要创建文本对象

1. 单击**绘图对象工具栏**上的 “文本”工具。
2. 在窗口内单击鼠标左键，并输入文本字符串。

   提示 要快速访问可应用于文本对象的各种命令，请右击文本对象，然后单击相应的命令。

要在文本字符串中显示数值

1. 单击“文本”工具，然后使用前面所述的有效数字格式之一在窗口中输入文本对象。例如：



2. 选择对象，然后在**特别**菜单上，单击**动画链接**或双击该文本对象。此时会出现**动画链接**选择对话框。

提示 要快速访问该对话框，请右击文本对象，然后单击**动画链接**。

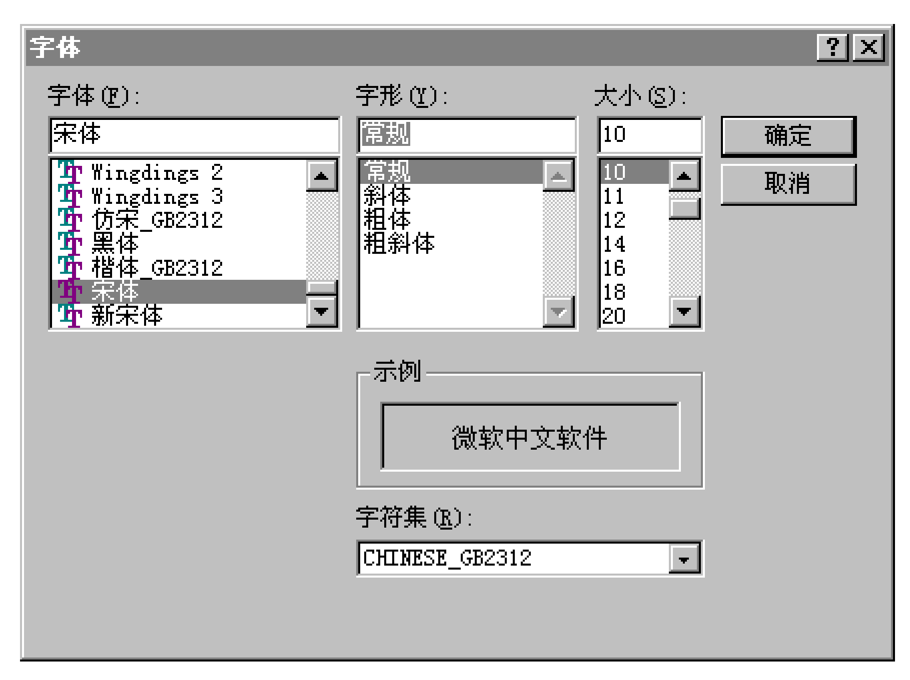
3. 在**值显示**区，单击**模拟**。此时会出现**输出 -> 模拟表达式**对话框。



4. 在**表达式**框中，输入一个模拟标记名或表达式。（在本例中，使用系统标记名 **$Second**）。
5. 单击**确定**。
6. Click the 单击菜单栏右上角的**运行时**快速切换开关 （或使用快捷键 ALT + !）切换到 WindowViewer，或者在**文件**菜单上，单击 **WindowViewer**。
7. 如果使用本示例，则会看到文本字符串中的磅 (#) 符号位置显示当前系统秒值 （在 0-59 之间）。
8. 单击菜单栏右上角的**开发**快速切换开关 （或使用快捷键 ALT + !）返回 WindowMaker，或者在**文件**菜单上，单击 **WindowMaker**。

要改变字符串的字体、字体样式和字体大小

1. 选择文本字符串，然后在**文本**菜单上，单击**字体**，或单击**格式工具**栏上的“字体”工具。此时会出现标准的 Windows **字体**对话框。



2. 从**字体**列表选择所需的字体（字体名出现在**字体**字段中）。选定字体之后，可供选择的样式和大小将出现在**字体样式**和**大小**字段中。在选定字体大小时，相应的样式和大小的字体样本会出现在**示例**字段中（详见上例）。
3. 单击**确定**。

备注 字体磅值会根据所选字体的可用磅值范围来放大或缩小。WindowMaker 的缺省字体是 System，并且不能改变大小。在改变大小之名，请选择 Windows True-Type 字体。

编辑文本对象

要改变对象中的文本

1. 选择包含文本的对象或按钮。
2. 在**特别**菜单上，单击**替换字符串**。此时会出现**替换字符串**对话框。

提示 要快速访问该对话框，请右击文本对象，指向**替换**，然后单击**替换字符串**。



如需有关 “模拟输出 / 显示”链接的详细信息，请参阅第 7 章 “创建动画链接”。

3. 在**新字符串**框中，输入新字符串，然后单击**确定**。

提示 您也可以给符号或单元包含的字符串使用此命令，以改变用“按钮”工具绘制的按钮上的标签。

在改变文本字符串时，它会保留原来的所有属性，包括字体、字形和颜色等。所有普通文本格式也适用于数值。

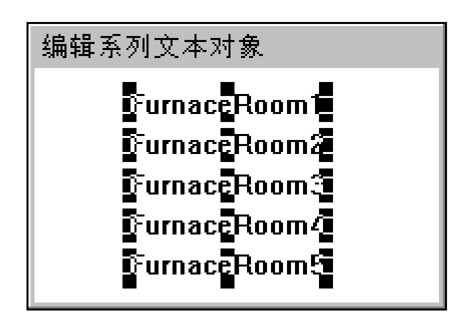
您也可以同时选择和编辑多个字符串对象。

替换文本对象的一部分

您可以改变文本对象的部分文本， InTouch 会自动对使用相同文本的所有选定文本对象作出更改。

要更改一系列文本对象中的部分文本

1. 选择所有的文本对象。



2. 在**特别**菜单上，单击**替换字符串**。此时会出现**替换字符串**对话框。

> **提示** 要快速访问该对话框，请右击文本对象，指向**替换**，然后单击**替换字符串**。

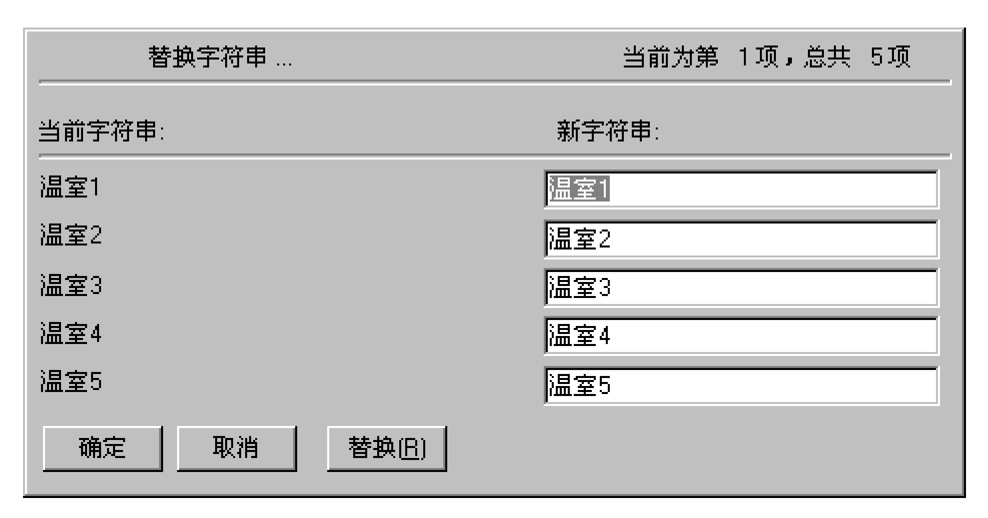


> **提示** 如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

3. 单击**替换**。此时会出现**替换文本**对话框。



4. 在**旧文本**框中，输入字符串中需要替换的部分。
5. 在**新文本**框中，输入替换文本。
6. 单击**确定**。此时会再次出现**替换字符串**对话框，显示对所选文本字符串所作的更改：



7. 单击**确定**。此时所有选定的文本对象均会自动修改。

使用线条和轮廓

您可以改变线条对象的样式和宽度，包含椭圆形、长方形、多边形、位图或图像的轮廓。您可以给单个或多个选定的对象应用线条样式或宽度改变。

线条菜单分成两个部分。上面的部分包含线条宽度宽，下面的部分包含线条样式。例如：



要应用线条命令

选择对象，然后在**线条**菜单上，单击所需的线条样式或宽度。

提示 如果在选择线条样式或宽度时未选择对象，则所作改变将应用于**向导工具栏**中所有线条工具的缺省设置。

备注 您只能改变实线的宽度。虚线只能是固定的一个像素宽。宽线条在运行时需要更长时间才能绘出。

要删除对象的轮廓

选择对象，然后在**线条**菜单上，单击**无线**。此时该对象的轮廓会被删除。

使用向导

在应用程序开发阶段，向导可以节省您大量的时间。它们易于使用且便于配置。要配置向导，请先进行安装，然后在**向导选择**对话框中选择它，将它粘贴到窗口中，然后双击它。此时会出现相应的配置对话框（假设可配置该向导）。

例如，如果要使用游标向导，则需要配置的项目包括标记名效果、游标的最大与最小范围标签以及填充颜色等等。通过使用向导，可以节省大量的开发时间，因为您不必再花时间单独去绘制对象的各个组件、设置对象的值范围，或设置对象的动画效果。

FactorySuite InControl 程序包括下列五个可放入 InTouch 窗口的向导。这些向导可实现 InControl 与 InTouch 之间简单而行之有效地交互。

向导	描述
InControl 项目	启动一个 InControl 项目。这会为特定的项目启动 InControl，供操作员使用所有的 InControl 功能，以编辑、编译、下载并运行该项目中的程序。
配置运行时引擎	启动运行时引擎并选择运行它的节点。
InControl 模式	用于将当前下载到运行时引擎的程序设置为特定的模式（运行、暂停、单步）。
InControl 编辑	启动项目中的单个程序。这会在开发环境下启动 InControl，并运行至指定程序中特定的行。您可以使用开发环境下所有可用的工具来编辑、编译、下载和运行程序。
InControl Runtime 添加标记	将 InTouch 标记名与 InControl 符号（变量）关联起来。

要安装或删除向导

1. 在**特别**菜单上，指向**配置**，然后单击**向导 /ActiveX 安装**，或者在“应用程序浏览器”中，双击**向导 /ActiveX 安装**。此时会出现**向导 /ActiveX 安装**对话框，并显示**向导安装**属性页。

提示 在 “应用程序浏览器” 中，您也可以右击**向导 /ActiveX 安装**，然后单击**打开**。



备注 同时最多可安装 43 个 “向导”。

2. 在**已安装的向导**列表中，选择要从应用程序中删除的向导，然后单击**删除**。此时会出现一个消息框，要求您确认删除。

备注 仅当向导显示在 “已安装的向导” 列表中时，**删除**按钮才会被激活。

提示 要选择一组向导，请单击列表中的第一个向导，在按住 shift 键的同时单击要选择的最后一个向导。如此将选定列表中第一个和最后一个选项之间的所有向导。要选择列表中的多个不连续向导，单击第一个向导，在按住 ctrl 键的同时单击下一个向导。对于要选择的每个向导，重复此步骤。

3. 单击**是**以删除向导。被删除的向导会移至**已卸载的向导列表**中。

备注 移除向导时，它并没有被删除。不过，它将不能再载入内存。

4. 要安装向导，请在**已卸载的向导列表**中选定向导，然后单击**安装**。

备注 仅当向导显示在**已卸载的向导列表**中时，**安装**按钮才会被激活。

5. 如果要从另一个目录安装向导，请单击**搜索**。此时会出现**搜索向导文件**对话框。

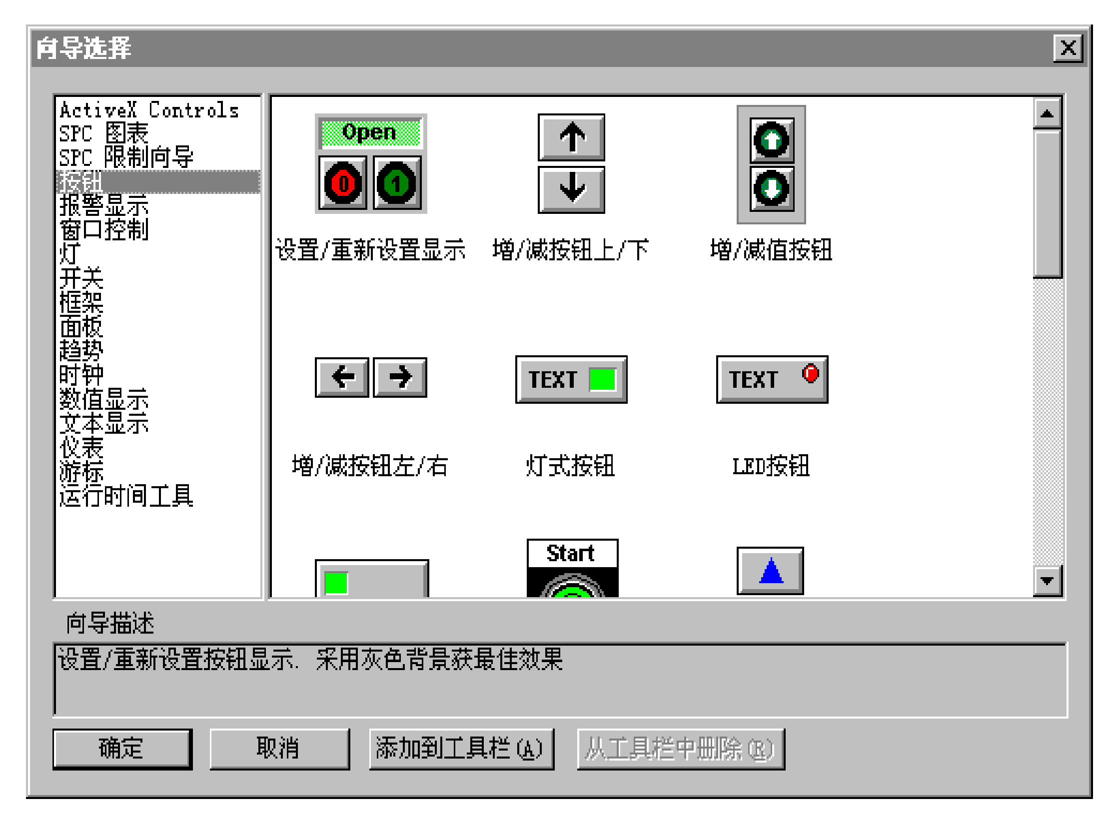


6. 找到包含要安装的向导的目录，然后单击**确定**。此时会再次出现向导安装对话框。

   找到的所有向导会出现在**已卸载的向导列表**中，您可以如前所述开始安装。

要将向导放入窗口

1. 单击**向导 /ActiveX** 工具栏上的 "向导对话框" 工具。此时会出现**向导选择**对话框。



2. 在向导列表中，单击要使用的向导类别。

   该类别中可用的所有向导均将出现在显示区中。例如，如果选择**按钮**，则所有可用的按钮向导会立刻出现在显示区。

3. 选择要使用的向导，然后单击**确定**或双击该向导。此时对话框关闭，您的窗口会重新出现。

提示 要将向导添加到**向导 /ActiveX** 工具栏中，请单击**添加到工具栏**。将向导添加到**向导 /ActiveX** 工具栏之后，您就可以随时选择该向导，并将它粘贴到打开的窗口中。

备注 可添加到工具栏中的向导数目取决于系统资源。

4. 返回窗口时，光标将变成弯头符号 。在窗口中单击要在该处粘贴向导的位置。

5. 双击向导以进行配置 （如果适用）。

备注 有些工具栏功能可用来直接修改适当的向导。例如，**字体缩小**工具、**线条颜色**工具以及**填充颜色**工具，等等。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章“WindowMaker 程序元素”。

要从工具栏删除向导

1. 单击**向导 /ActiveX 工具栏**上的 “向导对话框”工具。此时会出现**向导选择**对话框。
2. 单击**从工具栏中删除**。此时会出现**从工具栏删除向导**对话框。



3. 选择要从工具栏删除的向导。
4. 单击**确定**。

InTouch Windows 控件向导

Windows 控件向导是复杂的对象。与一般向导不同，它们通过 InTouch QuickScript 来提供增强功能。控件向导可用于编辑数据对象和操作员输入。 Windows 控件向导也具有 InTouch 标记名**点域**，并且在使用它们的 QuickScript 脚本函数时，可以在开发期间和运行时访问它们的一些属性。

您可以在 InTouch 应用程序中使用 Windows 控件向导，以便在运行时显示文本 / 数据、收集用户输入或向用户提供选项。选项形式可以是列表框、复选框、组合框和单选 （选项）钮。您可以使用文本框来显示或输入文本 / 数据。

在配置 Windows 控件向导时，您必须指定**控件名**来识别该控件。执行 Windows 控件 QuickScript 函数时， InTouch 将使用该**控件名**来识别控件。因此，您还必须在 QuickScript 函数中指定**控件名**参数。例如：

```
SetPropertyD ( "ControlName.Property", Discrete );
```

如需有关使用 Windows 控件 QuickScript 函数的详细信息，请参阅 *InTouch 参考指南*。

控件名不会计入应用程序的标记名计数，并且对于每个控件必须是唯一的。虽然对标记名不作要求，但它们对于有效使用控件却是必不可少的。例如，如果选定项目未自动指定给标记名，以使之可供 InTouch 访问，则在列表框中选择项目是没有意义的。

Windows 控件属性 （类似于标记名**点域**）可在开发时 (WindowMaker) 和运行时 (WindowViewer) 修改。它们也支持可在运行时处理的特定 QuickScript 函数，以修改列表、下载文件、停用控件，等等。

备注 为确保正常运行， Windows 控件对象不能互相重叠。为此，请在 WindowMaker 中选择对象，并确保该对象的手柄没有触到另一个对象。

接受键盘输入的 Windows 控件不能在 SuiteVoyager 下工作。

赋予列表框或组合框的标记名初始值不能用于初始化列表框或组合框的值。

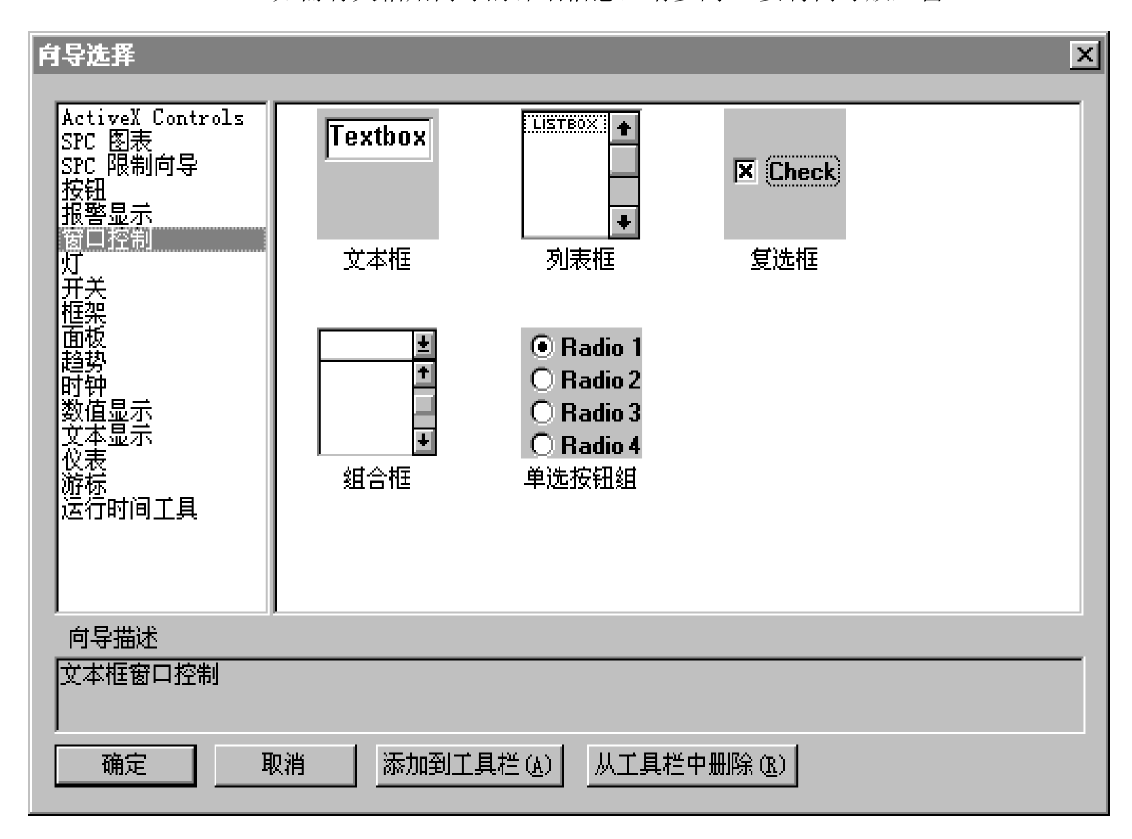
如需有关使用 Windows 控件**点域**的详细信息，请参阅 *InTouch 参考指南*。

使用 **InTouch Windows** 控件向导

可用的 Windows 控件向导包括：文本框、复选框、组合框、列表框、单选 （选项）钮等。 Windows 控件向导还具有标记名**点域**和 QuickScript 函数，您可以利用这些函数在运行时动态控制它们。

提示 与其它任何向导一样， Windows 控件向导可以粘贴到窗口中。

如需有关粘贴向导的详细信息，请参阅 “要将向导放入窗口”。



提示 要获取最佳的 Windows 控件三维效果，请给窗口选择灰色背景。如果窗口不能设置成灰色，请在 Windows 控件向导后面放置一个灰色的“窗格向导”。

Windows 控件使用准则

使用 Windows 控件向导时切记应遵循下列准则，这点非常重要：

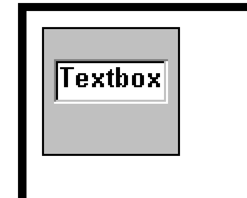
1. Windows 控件向导只有在不与其它窗口控件向导或其它普通图形对象重叠时，才能正常工作。

   提示 为确认 Windows 控件向导没有与其它任何对象重叠，请在 WindowMaker 中选定该向导。确认它的选择手柄没有与屏幕上的其它图形对象接触。

2. 使用 Windows 控件向导时务必谨慎小心。

   备注 在一个窗口中放置 10 到 20 个 Windows 控件向导不仅不美观，而且难以浏览显示。必须使用多个 Windows 控件向导时，我们建议您调用其它带 Windows 控件向导的对话框。

文本框控件向导

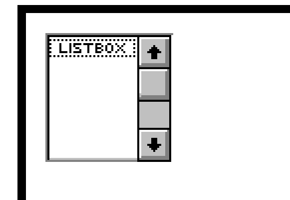
	文本框控件向导是一个多功能控件，可用于接受用户输入或显示文本，如“记事本”文件（仅限纯 ASCII 文件）。您可以配置文本框使之可以接受用户输入，或设置为只读以便仅用于显示。您只能给文本框控件向导指定**消息**型标记名。

文本框控件向导的 QuickScript 示例：

```
wcLoadText("TextBox_1",FileName);
wcSaveText("TextBox_1",FileName);
```

备注 如果标记名定义为最大长度，则只能从文本框内容将该字符数赋值给标记名。如果未给文本框指定标记名，则它的内容最长可达 65,535 个字符。

列表框控件向导

	列表框控件向导向用户显示一个选项列表。选项垂直显示在单个列中。如果项目数超过列表框中可显示的数目，则控件上会自动出现滚动条。列表框要求用户从列表中进行选择而不允许用户输入。您只能给列表框控件向导指定**消息**型标记名。

列表框控件向导 QuickScript 示例：

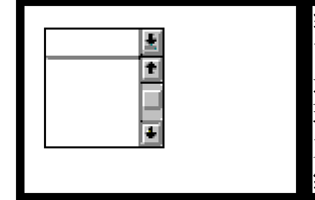
```
IF (ItemToAdd == "") THEN
    Show "Cannot Add Blank";
ELSE
    wcAddItem("ListBox_1",ItemToAdd);
{ 获取刚添加的项目的索引。 }
{ 因为列表已排序，我们不能对新项目的位置作任何假设。 }
    GetPropertyI("ListBox_1.NewIndex",ListBox_NewIndex);
{ 现在，设置由用户在屏幕上指定的 “项目数据”。 }
{ 从现在起，此项目将与此数据关联。 }
{ 它允许您将数字与字符串关联；字符串显示在列表中。 }
    wcSetItemData("ListBox_1",ListBox_NewIndex,
        ListBox_ItemData);
{ 由于我们刚添加项目，所以需更新 "NumItems" 变量。 }
    GetPropertyI("ListBox_1.ListCount",ListBox_NumItems);
ENDIF;
```

列表框和组合框控件向导使用内部初值为 1 的编号系统 （项目索引），它会给列表中的每个项目自动指定一个编号。例如，列表中的第一项编号为 1，第二项编号为 2，以此类推。项目索引是一个 32 位整数，用作 Windows 控件 “项目”的一个参数。

备注 对列表框和组合框使用 **wcLoadList()** 和 **wcSaveList()** 时，必须提供特定的格式和信息。

如需有关 Windows 控件 QuickScript 函数的详细信息，请参阅 *InTouch 参考指南*。

组合框控件向导

	组合框控件向导结合了文本框和列表框的功能。选项垂直显示在单个列中。如果项目数超过列表框中可显示的数目，则控件上会自动出现滚动条。组合框控件向导允许用户通过输入文本或从列表中选择项目来进行选择。您只能给组合框控件向导指定**消息**型标记名。

组合框有三种类型：

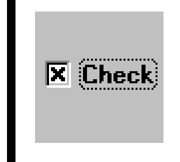
类型	描述
简单	组合框只显示列表。要显示列表框中的项目，列表框必须绘制得大到足以显示所有项目。如果项目太多而无法全部显示，则会自动插入一个垂直滚动条。简单组合框可供用户输入列表中未列出的选项，或显示与所输入字母匹配的列表中的第一项。如果未找到匹配项，则显示列表的顶部。
下拉	此组合框可供用户直接输入文本，或单击箭头来打开选项列表。与简单组合框一样，此控件允许用户输入列表中未列出的选项，或显示与所输入字母匹配的列表中的第一项。如果未找到匹配项，则显示列表的顶部。
下拉列表	此组合框类似于简单列表框。它们显示可供选择的选项列表。不过，与列表框不同的是，列表仅在单击箭头之后才显示。此类控件可用于节省屏幕空间。

组合框控件向导的 QuickScript 示例：

```
wcAddItem("ComboBox_1",UserMessage);
```

其中：*UserMessage* 是赋给字符串输入链接的标记名。操作员输入一则新消息，然后单击链接到**按下时** QuickScript 的这个动作按钮时，该消息会显示在控件名为 "ComboBox_1" 的组合框向导中。

复选框控件向导



复选框指出某个条件是开 / 关、真 / 假或是 / 否。复选框彼此独立，允许用户同时选择或取消选择任意多个复选框。复选框返回一个离散值。如果未选定，它们会返回 0 ；如果选定，则返回 1。您只能给复选框控件向导指定**离散**型标记名。

复选框控件向导的 QuickScript 示例：

```
{ 清除任何以前的机器 }
Machine = "";
IF (Cutter_Selected) THEN
   Machine = Machine + "Cutter";
ENDIF;
IF (Mixer_Selected) THEN
   Machine = Machine + "Mixer";
ENDIF;
```

其中：*Cutter_Selected* 是赋给第一个复选框向导中控件名 "Checkbox_1" 的标记名。 *Mixer_Selected* 是赋给第二个复选框向导中的控件名 "Checkbox_2" 的标记名。

Machine 是赋给用于显示选定复选框名称的字符串输出链接的标记名。

单选 （选项）钮控件向导



单选或选项钮提供一组选项供用户选择。与复选框不同，单选钮相互关联。选择一个单选钮会立即清除同组中其它所有的按钮。单选钮会返回一个整型值。单选钮控件的值对应于选定的单选钮。

例如，如果选定 Radio 1 选项，则当前值为 1 ；如果选定 Radio 4 选项，则值为 4。您只能给单选钮控件向导指定**整型**标记名。

单选 （选项）钮控件向导的 QuickScript 示例：

```
SelectedMachine=1;
```

其中：*SelectedMachine* 是赋予控件 "RadioButtonGroup1" 的单选钮向导的整型标记名。这是一个窗口**显示时** QuickScript，它将 *SelectedMachine* 标记名的值设置为 1。（这会将缺省值设置为初次显示窗口时选择组中的第一个单选钮）。操作员选择另一个单选钮时， *SelectedMachine* 的值会根据所选按钮而改变。例如，如果单选钮组有 4 个选项而操作员选择了第三个按钮，则 *SelectedMachine* 的值会设置为 3。

配置 Windows 控件向导

每个 Windows 控件向导根据其预期功能都具备独特的配置对话框。对话框中显示的选项都是可配置的属性，它们不能通过其它 WindowMaker 工具提供。不过，象颜色、字体样式和大小这样的属性可以通过相应的 WindowMaker 工具进行修改。

大多数 Windows 控件向导都支持 QuickScript 函数。例如，您可以创建一个 “数据改变” QuickScript 来加载和清除列表、在列表中添加和删除项目等。

如需有关 Windows 控件 QuickScript 函数及**点域**的详细信息，请参阅 *InTouch 参考指南*。

要配置 Windows 控件向导

1. 将向导粘贴到窗口中。

2. 双击它。此时会出现它对应的配置对话框。例如：

组合框控制
控制名 (C)： ComboBox_1
标记名 (T)：
确定
取消
属性
允许输入 (E)
可见 (V)
排序项目 (S)
允许重复项 (A)
最大长度 131
样式
简单 (P)
下拉 (D)
下拉列表 (L)

提示 如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

---

3. 在**控件名**框中，输入用于识别该 Windows 控件的唯一名称。

   为确保运行时正确操作，您必须给每个 Windows 控件向导指定一个唯一的**控件名**。执行 Windows 控件 QuickScript 函数时，WindowViewer 将使用该**控件名**来识别控件。**控件名**必须以字母字符开头 （第一个字母字符后可使用下划线或数字），并且不能包含任何特殊字符。

4. 您可以选择在配置 Windows 控件时指定**标记名**。不过，如果您确实指定标记名，则它的值会自动设置为控件的 **.Value** 属性 （即列表框中选定项目的索引）。

5. 输入所有其它必要的项目，并为当前配置的特定 Windows 控件设置所有适用的参数。

6. 单击**确定**。

Windows 控件向导属性

Windows 控件向导具有与 InTouch 标记名**点域**类似的属性。它们可以是可读写，也可以是只读的。有些属性可以在开发时访问，而另外一些则可以在运行时访问。它们可以标识为 *ControlName.x*，其中 *x* 指的是属性。

例如，如果 Windows 控件的 **.Visible** 属性等于零，则此控件在窗口中是不可见的。与 InTouch 标记名相似， **.Value** 是 Windows 控件向导的缺省属性。

在 WindowMaker 中， Windows 控件向导属性 （如文本字体、大小和颜色）均可通过使用相应的 WindowMaker 工具栏或菜单命令进行修改。那些工具栏或菜单命令不支持的属性则可以在向导的配置对话框中进行配置。 Windows 控件向导的其它属性是动态的，在运行时为可读写或只读。这类似于 InTouch 标记名 （**点域**）的运行时属性，如 **.Value** 和 **.Name**。与 InTouch 标记名不同的是， Windows 控件向导的运行时属性可通过 QuickScript 函数而不是动画链接表达式来访问。

根据属性的不同，运行时属性可以是可读写或只读。要控制或检索这些属性，必须使用 **GetProperty()** 和 **SetProperty()** QuickScript 函数。下面简要介绍每种 Windows 控件属性。

属性	描述
.Caption	确定复选框显示的 “消息”。
.Enabled	确定控件对象是否可响应操作员生成的事件。
.ListCount	确定列表框或组合框中的项目数。
.ListIndex	确定列表中当前选定项目对应的索引 （*标记名*或*数字*）。 **备注** 索引是定义列表中特定项目的编号。使用列表框时，索引 -1 表示当前没有选定任何项目。使用组合框时，索引 -1 表示用户已在控件的文本输入字段中输入新的文本。
.NewIndex	返回通过 **wcAddItem()** 或 **wcInsertItem()** 函数添加到列表框或组合框的最后一个项目对应的整型索引 （*标记名*）。
.ReadOnly	确定文本框的内容是只读还是可读写。
.TopIndex	确定列表框最顶端项目对应的整型索引。
.Value	所有 InTouch Windows 控件向导的缺省属性。此属性发生的改变在 InTouch 标记名和 Windows 控件向导中保持同步。
.Visible	确定 Windows 控件在窗口中是否可见。 Windows 控件向导属性不会出现在**选择域名**对话框中。

例如：

```
[ErrorNumber=]GetPropertyM("ControlName.Property",Tagname)
    ;
```

其中：

参数	描述
ControlName	为 Windows 控件向导配置的**控件名**，例如 CheckBox_1，或报警对象的名称，例如 AlmObj_1。

参数	描述
.Property	Windows 控件或报警对象的属性。
标记名	有效的 InTouch 标记名 （与要返回的值类型相同）， 用于存放处理函数时的属性值。

如需有关 Windows 控件向导的详细信息，请参阅 “InTouch Windows 控件向导”。"

Windows 控件向导函数

下面简要介绍可在 InTouch Windows 控件向导中使用的 InTouch QuickScript 函数：

函数	描述
wcAddItem	将提供的字符串添加到列表框或组合框中。
wcClear	从列表框或组合框删除所有的项目。
wcDeleteItem	删除与列表框或组合框中的项目索引参数关联的项目。
wcDeleteSelection	从列表中删除当前选定的项目。适用于列表框和组合框
wcErrorMessage	对于给定的错误号， wcErrorMessage() 会返回描述该错误的字符串消息。适用于列表框、文本框、组合框、单选钮和复选框。
wcFindItem	确定列表框或组合框中与提供的字符串相符的第一个项目所对应的索引。
wcGetItem	返回与列表框或组合框中相应的索引关联的项目字符串的值属性。
wcGetItemData	检索与列表框或组合框中的列表项关联的整型值。
wcInsertItem	将字符串插入列表框或组合框。
WcLoadlist	使用新项目替换列表框或组合框的内容。
wcLoadText	使用新字符串替换文本框的内容。
wcSavelist	使用列表对象中的项目替换文件名的内容。
wcSaveText	将文本框包含的文本保存到文件名。
wcSetItemData	将一个整数值指定给列表框中的某个项目。

使用 **ActiveX** 控件

ActiveX 控件的前身是 OLE 控件或 OCX，它们是以标准方式执行特定功能的独立软件组件。 ActiveX 控件为可重复使用的组件定义了标准接口。 ActiveX 控件不是独立的应用程序。相反，它们是放置到控件容器中的服务器。要使用 ActiveX 控件，必须将它放入 ActiveX 容器。 InTouch 便是一个 ActiveX 容器。 Microsoft VisualBasic 和 Internet 浏览器也是 ActiveX 容器。

ActiveX 控件与 InTouch 向导几乎完全一样，不同的是它给 InTouch 应用程序带来了引人注目的新功能。您可以使用 Visual Basic、 Microsoft VC++ 或其它第三方开发工具来创建 ActiveX 控件。您也可以从第三方购买某些特殊功能的 ActiveX 控件。这些控件以 OCX 的形式封装。Wonderware 的 FactorySuite InTrack 组件也提供了多种 ActiveX 控件。此外，IndustrialSQL 的 ActiveTrend 可供您在 InTouch 内运行 IndustrialSQL Trend 程序 （或函数子集）， ActiveEvent 可供您在另一个应用程序中发生某个事件时通知 IndustrialSQL “事件”子系统。

ActiveX 控件有三个主要组成部分：*属性*、*方法*与*事件*。

- 属性非常类似于可以修改的变量，例如 Calendar.day、 Control.height 等。
- 方法则类似于可以从容器调用的脚本函数调用。例如，**Browser.Navigate("http://www.wonderware.com")**、 **Engine.start()**。
- 事件通过 ActiveX 容器发生。例如， **Control.click (shift)**。**FileViewer.DoubleClick (name)** 等等。

InTouch 允许您访问 ActiveX 控件属性、方法和事件。您可以将这些属性与 InTouch 标记名关联，或通过编写 InTouch QuickScript 来访问它们。

备注 为了让 “ActiveX 事件”脚本正常工作，为之创建脚本的 ActiveX 控件必须载入内存。如果关闭了包含 ActiveX 控件的窗口，则它的 “ActiveX 事件”脚本，或包含与该 ActiveX 控件关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

您可以在 InTouch 应用程序中使用一个或多个 ActiveX 控件。 InTouch 允许您方便地选择 ActiveX 控件、将它们粘贴到任何应用程序窗口，以及将它们添加到**向导 /ActiveX 工具栏**。您也可以将 “ActiveX 事件”脚本从一个应用程序导入另一个应用程序。

在 InTouch 中使用 ActiveX 控件

1. 安装要使用的 ActiveX 控件。
2. 选择该 ActiveX 控件并将它粘贴到 WindowMaker 窗口。
3. 配置 ActiveX 控件的属性并将它们指定给标记名。
4. 将 ActiveX 事件与 “ActiveX 事件”脚本关联起来。
5. 调用 ActiveX 方法，并在 “ActiveX 事件”脚本或其它 InTouch QuickScript 中设置 ActiveX 控件的属性。

您可以在 WindowMaker 中对 ActiveX 控件执行下列编辑操作：

- 调整 ActiveX 控件的大小 （如果该控件支持调整大小）。
- 复制、剪切、拷贝、粘贴和删除 ActiveX 控件。
- 对 ActiveX 控件使用所有的对齐命令 （左、右、顶、底、中心点对齐）。
- 将 ActiveX 控件添加到**向导 /ActiveX 工具栏**。
- 在创建单元时将 ActiveX 控件和其它对象包含到一起。

- 使用 WindowMaker 菜单命令及对应的工具栏工具去直接修改许多 ActiveX 属性。例如，字体缩小、线条颜色、填充颜色等。

InTouch 不支持下列类型的 ActiveX 控件：

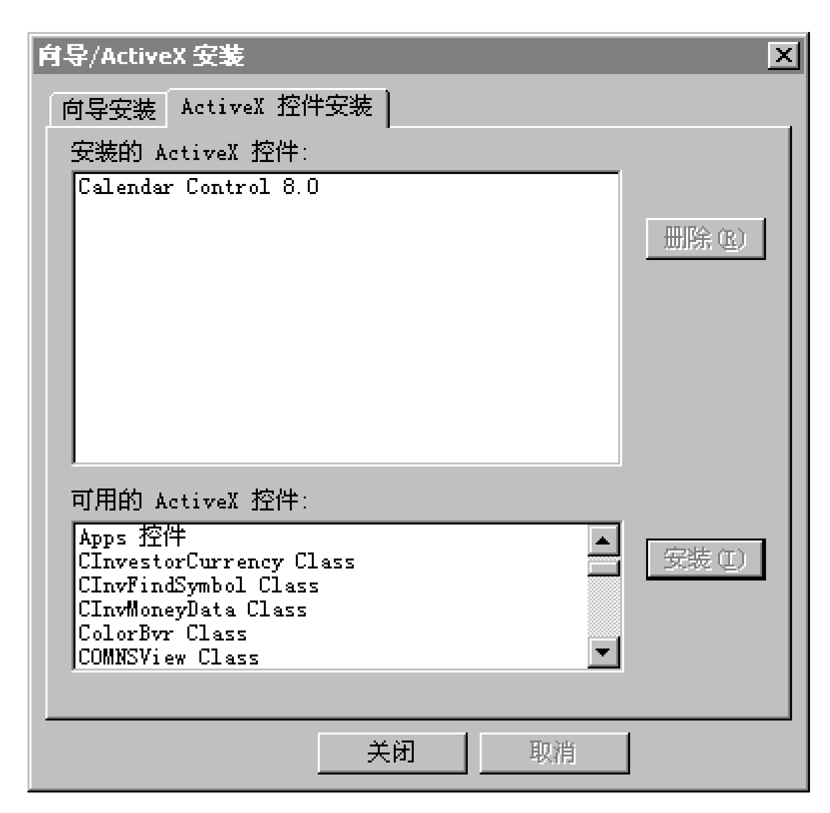
- 无窗口控件
- 单框架布置 （组合框）
- 容器
- 数据控件
- 发送对象
- Array、 Blob、 Object、 Variant 型控件

要安装或删除 ActiveX 控件

1. 在**特别**菜单上，指向**配置**，然后单击**向导 /ActiveX 安装**，或者在 “应用程序浏览器”中，双击**向导 /ActiveX 安装**。此时会出现**向导 /ActiveX 安装**对话框。

提示 在 “应用程序浏览器”中，您还也可以右击**向导 /ActiveX 安装**，然后单击**打开**。

2. 单击 **ActiveX 控件安装**选项卡以激活 **ActiveX 控件安装**属性页：



3. 在**已安装的 ActiveX 控件**列表中，选择要从应用程序中删除的控件，然后单击**删除**。此时会出现一个交互式消息框，要求您确认删除。

提示 要选择一组控件，请单击要选择的第一个控件，在按住 **Shift** 键的同时，选择所需的最后一个控件。如此将选定介乎其间的所有控件。要选择列表中不连续的多个控件，请单击第一个控件，在按住 **Ctrl** 键的同时，单击下一个控件。

4. 单击**是**以删除控件。被删除的控件将移至**可用的 ActiveX 控件**列表中。

备注 删除控件时，它并不会真的被删除。不过，它将不能再载入内存。也就是说，它将不能再正常工作。

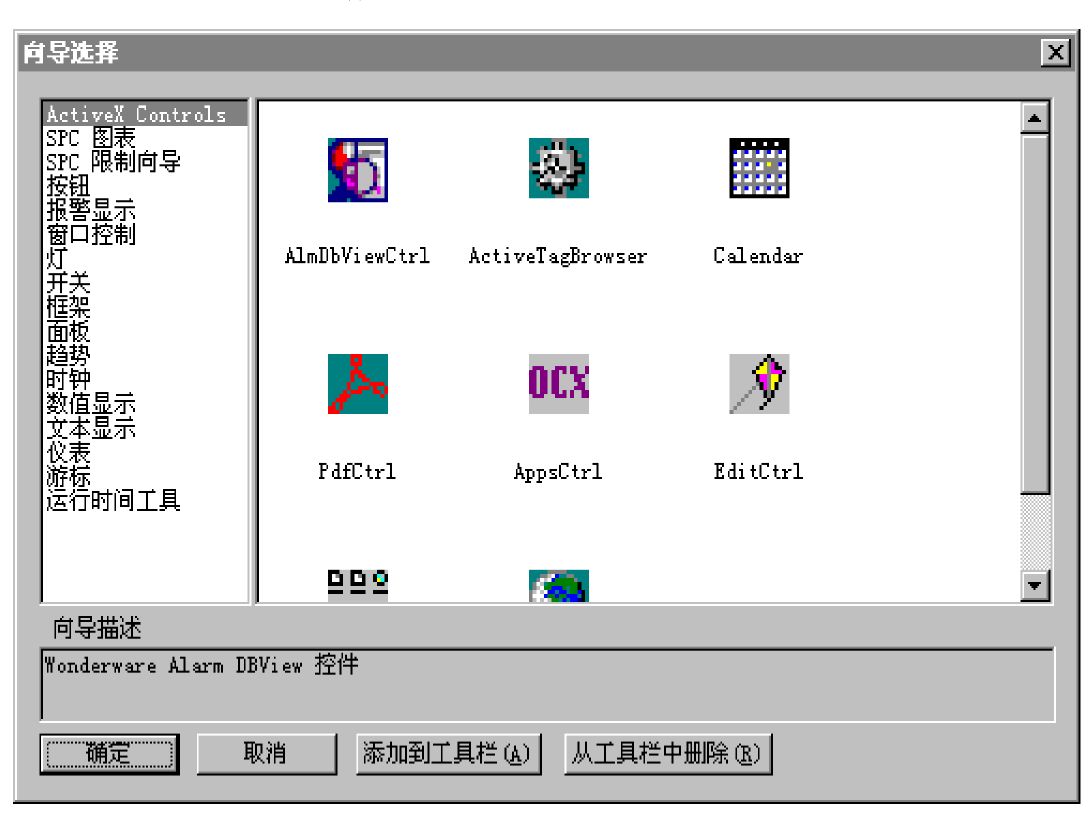
5. 要安装 ActiveX 控件，请在**可用的 ActiveX 控件**列表中选择它们，然后单击**安装**。

备注 **安装**按钮只有当**可用的 ActiveX 控件**列表中显示有控件时才会变为活动状态。

6. 单击**关闭**。

要将 ActiveX 控件放入窗口

1. 单击**向导 /ActiveX** 工具栏上的"向导对话框"工具。此时会出现**向导选择**对话框。



2. 在向导列表中，单击 **ActiveX 控件**类别。此时所有可用的 ActiveX 控件均出现在显示区中。

3. 选择要使用的 ActiveX 控件，然后单击**确定**，或者双击该控件。此时对话框关闭，您的窗口会重新出现。

提示 要将 ActiveX 控件添加到**向导 /ActiveX 工具栏**，请单击**添加到工具栏**。将控件添加到**向导 /ActiveX 工具栏**之后，您就可以随时选择该向导，并将它粘贴到打开的窗口中。

备注 可添加到工具栏中的 ActiveX 控件数取决于系统资源。

4. 返回窗口时，光标会变成弯头符号 。单击窗口中要粘贴 ActiveX 控件的位置。
5. 双击控件以配置其属性。

如需有关 WindowMaker 工具栏的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

要从工具栏中删除 ActiveX 控件

1. 单击**向导 /ActiveX 工具栏**在的**向导对话框**工具。此时会出现**向导选择**对话框。
2. 单击**从工具栏中删除**。此时会出现**从工具栏中删除向导**对话框。



3. 选择要从工具栏中删除的 ActiveX 控件。
4. 单击**确定**。

配置 ActiveX 控件

将 ActiveX 控件粘贴到 InTouch 窗口时，您必须配置它的属性以便与 InTouch 进行交互。您必须给每个控件命名，以便可以从 InTouch QuickScript 中引用。粘贴 ActiveX 控件时，系统会生成一个缺省控件名，如 Calendar1。（此控件名在 InTouch 应用程序中是全局性的）。

您必须将 ActiveX 控件的属性指定给 InTouch 标记名。每种属性类型必须指定给相应的 InTouch 标记名类型。

要给 ActiveX 控件命名

1. 将 ActiveX 控件粘贴到 WindowMaker 窗口中。
2. 双击该控件，或右击该控件，然后单击**属性**。此时会出现该控件对应的**属性**对话框。

备注 每个 ActiveX 控件的 “属性”对话框对于控件而言是唯一的。显示的选项卡数取决于特定控件的属性。某些 ActiveX 控件可能要求您配置更多的属性。例如某些控件可能要求您配置它们的 “颜色”和 “字体”，而其它一些控件则可能没有这些属性。不过，对于所有的 ActiveX 控件， InTouch 均包括三个选项卡：**控件名**、**属性**和**事件**。例如：



3. 单击**控件名**选项卡，然后在**控件名**框中给 ActiveX 控件输入一个唯一的名称。

   您必须给 InTouch 应用程序中使用的每个 ActiveX 控件定义一个唯一的名称。脚本函数使用 “控件名”来识别控件。例如：

   ```
   #Calendar1.day = Tag1;
   #Calendar1.year = 1997
   ```

备注 在缺省条件下，“控件名”由该控件的 ProgID 确定。 ProgID 是在计算机中安装 ActiveX 控件时输入系统注册表的名称。将该控件的实例放入 InTouch 应用程序时，程序会从系统注册表读取该控件的 ProgID，并在其后附加一个索引号，结果就生成了控件名，如 **Calendar1**。

如果使用缺省的 “控件名”，则程序会在下列情况下创建控件的一个新实例，并为它指定一个唯一名称：

- 在**编辑**菜单上选择**复制**
- 在**编辑**菜单上选择**剪切**或**复制**，然后选择**粘贴**
- 在**文件**菜单上选择**窗口另存为**

- 单击**撤消**，然后单击**恢复**
- 导入包含控件的窗口

改变 ActiveX 控件名

您最好不要改变 ActiveX 控件的名称，但在某些特殊情况下您可能需要改变控件名。假定使用缺省 “控件名”，例如 **Calendar1**。后来您删除了该 ActiveX 控件，然后又重新创建它。InTouch 会自动给 “控件名” 后面的序号加 1。在这种情况下，重复的 ActiveX 控件的名称会是 **Calendar2**。为了让您现有的脚本能够识别这个新控件，您必须将新控件从 **Calendar2** 重命名为 **Calendar1**。

ActiveX 控件必须具有唯一的名称。编辑 “控件名” 并单击**确定**或**应用**时，该控件名会与现有的 “控件名” 表进行比较。如果表中不存在该名称，则控件名会改变并得以保存。如果已经存在该名称，则会出现一则错误消息。此时，您必须为该控件指定一个唯一的名称。

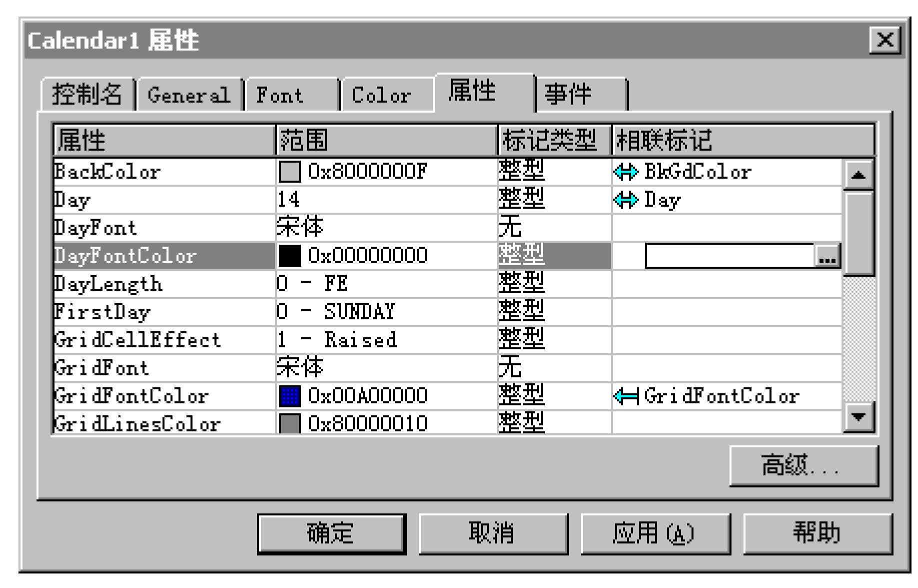
配置 ActiveX 控件属性

可以给特定的 ActiveX 控件配置的属性是由 ActiveX 控件的设计人员确定的。每个 ActiveX 控件的**属性**属性页会显示三列：**属性**、**范围**及**关联标记**。**属性**和**范围**列是只读的。**关联标记**列用于将 InTouch 标记名与**属性**列中相应的属性进行关联。

备注 单击**范围**列中的某个项目时，会出现一个箭头，供您单击它来浏览此项目的可能值列表。列表区中的项目只能查看而不能更改。

要配置 ActiveX 控件的属性

1. 单击 ActiveX 控件**属性**对话框中的**属性**选项卡，以激活**属性**属性页：



2. 单击**关联标记**列中每个单元格的中间，为相应的属性输入标记名。

提示 如果输入了一个“标记名字典”中尚未定义的标记名，则系统会提示您立即去定义它。

如果双击空白单元格，或者单击 [...] 按钮，则会出现“标记浏览器”，显示所选标记源的标记名。双击要使用的标记名，或选择它然后单击**确定**。此时该标记名会自动插入单元格。

备注 初始化标记名与创建 ActiveX 控件之间会出现定时问题。由于此定时问题，很难保证标记名的初始值等于 ActiveX 控件中相关属性的值。

为解决此问题，必须在“启动”事件与 InTouch QuickScript 之间建立一种联系。在 QuickScript 中，可采用适当的逻辑将标记名值设置为控件的属性。这可以通过使用赋值语句来实现。例如：

```
#ThisControl.Property = MyTag;
```

事件发生之后， QuickScript 即开始执行，控件的属性与标记名的值将保持同步。因此，属性与标记名之间会出现正常的通知。

如需有关“标记浏览器”的详细信息，请参阅第 6 章“标记名字典”。

3. 指定标记名之后，右击标记名左边单元格以选择标记名与相应属性的关联方向。（连续双击可在各种关联方向选项之间进行切换。关联方向选项将在下文叙述）。

提示 **关联标记**列的每个单元格实际包含两个字段。“关联方向选择”和“标记名项目”。 ActiveX 控件确定关联方向，属性类型确定必须使用的标记名类型。

您可以选择单向或双向关联。不过，如果选择的关联方向对此属性或标记名而言无效，则该控件自动作相应改变。例如，如果选择 ⇐，则在标记名的值发生改变时，它关联的属性也会相应改变。根据属性和标记名之间的潜在联系，会出现下列符号的特定子集。

例如，如果将属性与可写标记名关联，您看到的关联子集会合将相同的属性与只读标记名关联时的不同。选择相应的关联符号，具体如下所示：

符号	描述
⇐	标记名设置关联属性的值。
⊢	此符号表示该属性为只读，因此标记名不能改变属性的值。
⇒	属性设置关联标记名的值。
⊣	此符号表示标记名为只读，因此属性不能改变标记名的值。
⇔	可从标记名或属性去设置值。（标记名优先）。

⟷	标记名和属性都是只读的。
⇐	标记名可改变属性的值，但属性不能改变标记名的值。属性不能改变标记名的值是因为属性不是绑定的，或者标记名是只读的。
⇒	属性可以改变标记名的值，但标记名不能改变属性的值。标记名不能改变属性的值是因为属性为只读。

4. 单击**确定**。

备注 您也可以通过 “ActiveX 事件”脚本和 （或）其它 InTouch QuickScript 来访问或改变属性。所有的 ActiveX 脚本函数均使用 #（磅）号注明。访问 ActiveX 属性的有效语法是：

```
#ControlName.PropertyName
```

例如：

```
#Calendar1.Day = 29;
Tag1 = #Calendar1.year;
```

如需有关详细信息，请参阅 “配置 ActiveX 控件”。

备注 更改 ActiveX 属性页内容时，无论是在属性页上单击 “应用”还是 “取消”，所作更改都会立即生效。在控件属性页上更新的属性可能不会反映在 InTouch ActiveX 属性页中。在选项卡之间来回切换会更新属性页。

InTouch 不支持 "SelectionChange" 事件。

使用 ActiveX 控件方法

ActiveX 控件方法类似于 ActiveX 控件属性。控件方法在运行时 (WindowViewer) 激活。 ActiveX 控件方法可以通过 “ActiveX 事件”脚本和 （或）其它 InTouch QuickScript 进行访问。

备注 为了让 “ActiveX 事件”脚本正常工作，为之创建脚本的 ActiveX 控件必须载入内存。如果关闭了包含 ActiveX 控件的窗口，则它的 “ActiveX 事件”脚本，或包含与该 ActiveX 控件关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

要使用 ActiveX 方法和（或）属性

1. 在 ActiveX 控件的**属性**对话框中，单击**事件**选项卡以激活**事件**属性页：



2. 双击**脚本**列中的空白单元格。此时会出现 **ActiveX 事件脚本**编辑器。



3. 在**插入**菜单上，单击 **ActiveX**。此时会出现 **ActiveX 控件浏览器**。

4. 在**控制名**列表中，选择要访问其方法或属性的 ActiveX 控件。

   应用程序当前使用的所有 ActiveX 控件均会列出。

备注 如果选择 **ThisControl** 而不是实际的 "控制名"，则显示的是应用程序中当前选定的 ActiveX 控件的方法或属性。通过选择 **ThisControl** 而不是实际的控件名，您可以创建一般的 "ActiveX 事件" 脚本函数。随后，您可以将这些函数复制和粘贴到任何其它 "ActiveX 事件" 脚本或其它 InTouch QuickScript 中，而不必在新的脚本中改变 "控件名"。例如：

```
#ThisControl.Navigate
("http:\\www.wonderware.com");
#ThisControl.Navigate(URL);   { 其中 URL 是一个标记名
}
```

ThisControl 只能通过 "ActiveX 事件" 脚本进行访问。它不能通过任何其它类型的 InTouch QuickScript 进行访问。

ThisControl 用于编写可在相同类型的控件的任何实例中重复使用的事件脚本的一般引用。选定 **ThisControl** 时，列表中显示的属性和方法与针对特定控件的列表中的相同。

---

5. 在**方法 / 属性**列表中，选择要在脚本中使用的方法或属性。

提示 方法是列表中包含括号的项目。例如 **Display()**。

---

6. 单击**完成**。此时选定的控件名和方法或属性会自动插入脚本中。

> **提示** 在所有其它类型的 InTouch QuickScript 中， ActiveX 控件的方法和属性也通过**插入**菜单进行访问。

使用 ActiveX 控件事件参数

通过设计特定的动作并将它与事件关联，您可以在运行时 (WindowViewer) 执行 ActiveX 控件事件。例如，如果 ActiveX 控件带有错误事件处理器，则您可以创建一个 “ActiveX 事件” QuickScript，在出错时显示一个包含错误消息的窗口。“ActiveX 事件”脚本可支持事件动作。You can associate a named event script to each event..

要使用 ActiveX 事件参数

1. 双击要为之创建 “ActiveX 事件”脚本的 ActiveX 控件。此时会出现所选 ActiveX 控件的**属性**对话框。
2. 单击**事件**选项卡以激活**事件**属性页：



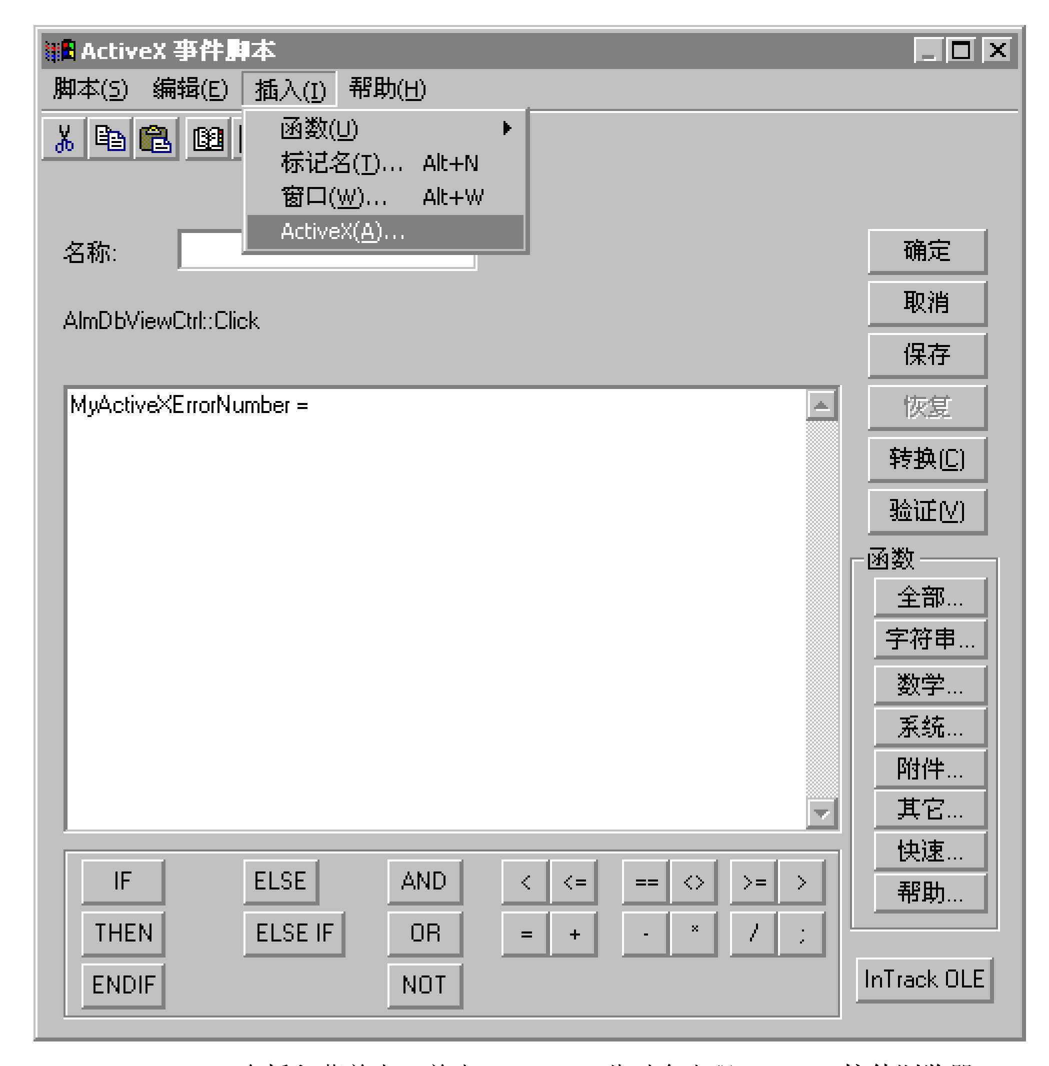
3. 在**事件**列中，选择要与 “ActiveX 事件”脚本关联的事件。
4. 在**脚本**列相应的单元格中，为要创建的 “ActiveX 事件”脚本输入一个唯一的名称，然后双击该名称，或单击**确定**。此时会出现下面的消息框。



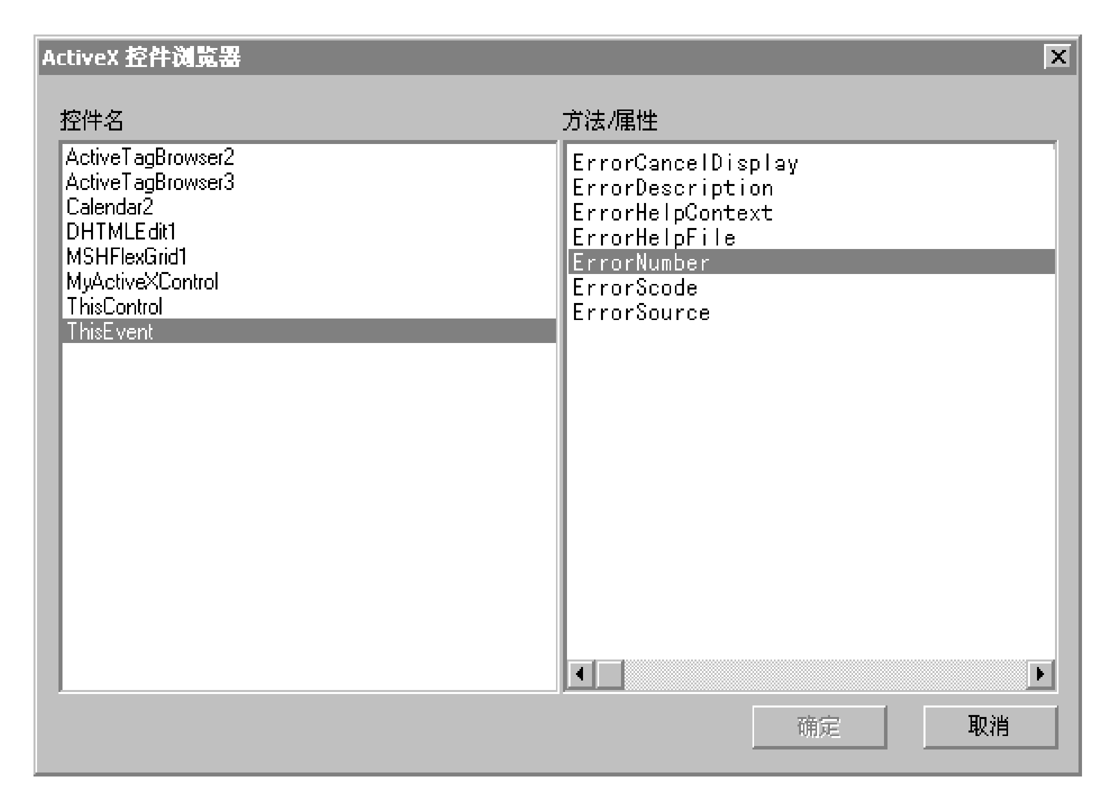
5. 单击**确定**。此时会出现“ActiveX 事件”脚本编辑器，显示在**名称**框中输入的名称（详见下例）。如果双击一个空白的**脚本**单元格，则在出现“ActiveX 事件”脚本编辑器时，您必须给“ActiveX 事件”脚本输入一个名称。

提示 如果要使用的“ActiveX 事件”脚本已经存在，请单击 [...] 按钮。此时会出现**选择 ActiveX 脚本**对话框，列出应用程序中现有的全部“ActiveX 事件”脚本。

如需有关详细信息，请参阅“重复使用 ActiveX 事件脚本”。



6. 在**插入**菜单上，单击 **ActiveX**。此时会出现 **ActiveX 控件浏览器**。



7. 在**控制名**列表中，选择 **This Event** 以访问所选事件的参数。在本例中，所选的事件是 **Error**。

备注 ThisEvent 只能通过 "ActiveX 事件" 脚本进行访问。它不能通过任何其它类型的 InTouch QuickScript 进行访问。您必须选择 **ThisEvent** 以访问 ActiveX 控件的事件参数。

事件参数包含有用的信息。例如，在出现按键事件后，您想要了解这样的问题："我们都知道按了键，但到底按的是哪个键呢"？如果此信息很重要，则控件的开发人员可能会添加一个事件参数来存储键代码。要访问此参数，您需要选择 **ThisEvent**，然后选择参数（可能会被称作 **keypressedkeycode**）。需要特别注意的是，大多数的 ActiveX 事件都没有参数。

事件不一定会在运行时传递参数。事件参数可以通过使用 **ThisEvent** 关键字进行访问。例如：

```
MyActiveXErrorNumber = #ThisEvent.ErrorNUmber;
```

其中： # indicates that this is an ActiveX script function.**ThisEvent** 与 ActiveX 控件的**事件**属性页中选定的事件关联， **ErrorNumber** 是由选定的事件传递的参数。

8. 在**方法 / 属性**列表中，选择要在 "ActiveX 事件" 脚本中使用的方法或属性。

9. 单击 “完成”。选定的控件名（在本例中是 "ThisEvent"）及选定的事件参数均会自动插入脚本中的光标所在位置。例如：



备注 “ActiveX 事件”脚本引用由下列几部分组成：

表示 “我是一个 ActiveX 脚本引用”的数字符 (#)。

控件的名称。

分隔符 (.)，将控件名与所调用的属性或方法分开。

属性、方法或事件参数。

10. 单击**确定**以保存 “ActiveX 事件”脚本并关闭脚本编辑器。此时会再次出现 ActiveX 控件的**属性**对话框。
11. 单击**确定**以关闭**属性**对话框，或者您可以继续创建 “ActiveX 事件”脚本。

重复使用 **ActiveX** 事件脚本

只有相同事件和相同类型的 ActiveX 控件才能重复使用 “ActiveX 事件” 脚本。例如，按下鼠标事件可能是数百个 ActiveX 控件中的一个库存事件。不过，在 ActiveX ControlA 中为按下鼠标所编写的 “ActiveX 事件” 脚本不能由 ActiveX ControlB 中的按下鼠标事件所重复使用，除非这两个控件属于相同的类型。

要重复使用 **ActiveX** 事件脚本

1. 双击要为之重复使用现有 “ActiveX 事件” 脚本的 ActiveX 控件。此时会出现所选 ActiveX 控件的**属性**对话框。

2. 单击**事件**选项卡以激活**事件**属性页：



3. 在相应事件的**脚本**列中，单击 按钮。此时会出现**选择 ActiveX 脚本**对话框。



此对话框将仅显示为相同类型 ActiveX 控件和相同选定事件所编写的 “ActiveX 事件”脚本。

例如，假定要为第二个 ActiveX Calendar 控件的 "Click" 事件创建一个 “ActiveX 事件”脚本。您已经在应用程序中创建了其它两个 “ActiveX 事件”脚本，分别是 Click1 和 Click2。 Click1 是为一个不同的 ActiveX Calendar 控件的 "Click" 事件创建的，而 Click2 是为一个 ActiveX InSQL Trend 控件的 "Click" 事件创建的。在单击 按钮并且出现**选择 ActiveX 脚本**对话框时，它将仅显示 Click1 的脚本，因为它是为相同的 ActiveX 控件和相同的事件而创建的。

4. 选择要使用的“ActiveX 事件”脚本，然后单击**确定**。

   所选脚本的名称会自动插入先前单击 [...] 按钮的**脚本**单元格中。例如：



5. 单击**确定**以关闭**属性**对话框，或继续创建“ActiveX 事件”脚本。

导入 ActiveX 事件脚本

将“ActiveX 事件”脚本从一个 InTouch 应用程序导入当前的应用程序，可以为您节省大量的开发时间。将“ActiveX 事件”脚本从一个 InTouch 应用程序移到另一个应用程序时，您必须使用 WindowMaker **文件**菜单上的**导入**命令。

备注 将“ActiveX 事件”脚本从一个应用程序导入另一个应用程序时，所有 ActiveX 事件脚本均会被导入。此外，为了让导入的“ActiveX 事件”脚本在新应用程序中能够正常运行，原先创建脚本时的相同 ActiveX 控件和相同的事件也必须用在新应用程序中，并且必须加载到内存中。如果关闭了包含 ActiveX 控件的窗口，则它的“ActiveX 事件”脚本，或包含与该 ActiveX 控件关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

如需有关导入脚本的详细信息，请参阅第 8 章“在 InTouch 中创建 QuickScript”。

自定义运行时环境

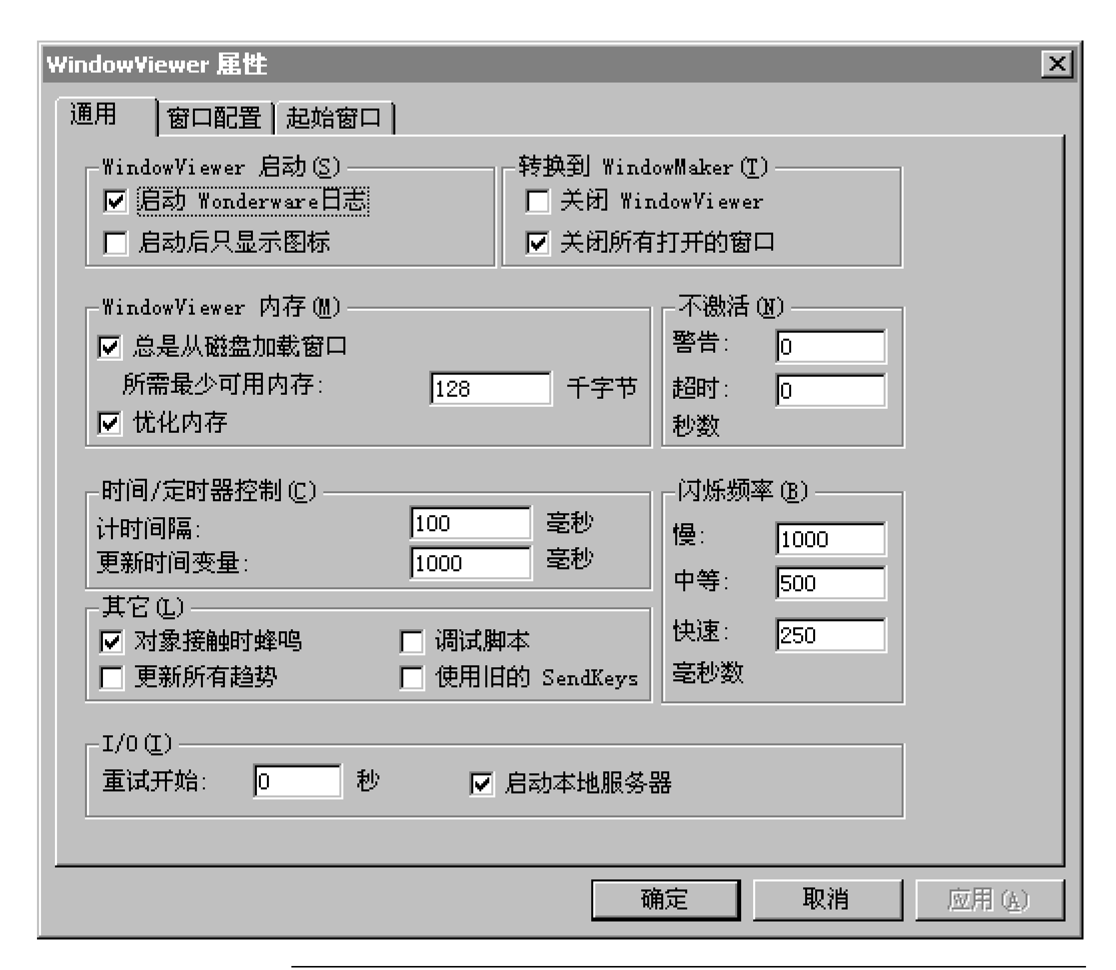
与 WindowMaker 类似，您可以设置许多属性来自定义运行时环境 (WindowViewer)。例如，您可以设置闪烁对象的闪烁速度、系统闲置超时及警告值、在 WindowViewer 从它的图标或菜单命令启动时自动打开的窗口，等等。

设置 WindowViewer 的通用属性

要设置 WindowViewer 的属性

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**，或者在“应用程序浏览器”中的**配置**下，双击 **WindowViewer**。此时会出现 **WindowViewer 属性**对话框，并显示**通用**属性页。

提示 在“应用程序浏览器”中，您也可以右击 **WindowViewer**，然后单击**打开**。



提示 如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

---

2. 如果希望 WindowViewer 作为一个图标而不是窗口启动，请选择**启动后只显示图标**。

提示 仅当您使用 WindowViewer 为其它 I/O 互连应用程序收集数据时，才可选择此选项。

---

3. 如果希望在启动 WindowMaker 时自动关闭 WindowViewer，请选择**关闭 WindowViewer**。

提示 如果内存足够，并且您使用快速切换开关在 WindowViewer 和 WindowMaker 之间转换，请不要选择此选项。

快速切换选项在 **WindowMaker 属性 - 通用**对话框中选定。

4. 选择此选项时， **WindowMaker 属性 / 通用**属性页上的**转换到 WindowViewer 时关闭**选项也将自动选定。
5. 如果希望在从 WindowViewer 转换到 WindowMaker 时自动关闭所有打开的窗口，请选择**关闭所有打开的窗口**。

提示 选择此选项可以释放系统内存。

6. 如果内存不足，请选择**总是从磁盘加载窗口**。

提示 选择此选项，应用程序窗口将从磁盘加载，并且在关闭时不会保存到 RAM 存储器。

7. 在**所需最少可用内存**框中，输入要为其它 Windows 应用程序保留的可用内存千字节数。
8. 选择**优化内存**以显著加快图形的更新速率。选择此选项也将明显提高文本字段的更新速率。

提示 如果系统内存不足，请不要启用此选项。

9. 在**警告**框中，输入在系统离散型标记名 **$InactivityWarning** 设置为 1 (True) 之前无操作员活动 （鼠标或击键）的时间 （以秒计）。

提示 您可以在“条件”QuickScript 中使用 **$InactivityWarning** 来显示一个窗口，以警告操作员即将从系统注销。如果操作员在指定的超时范围内，单击鼠标、击键或者使用其它定向设备执行任一动作，则它将不会注销。 **$InactivityWarning** 与计时器均会重置。

10. 在**超时**框中，输入系统离散型标记名 **$InactivityTimeout** 设置为 1(True) 之前无操作员活动 （鼠标或击键等）的时间 （以秒计）。在 **$InactivityTimeout** 为真时，系统会将登录的操作员用户名视为保留名 "None"，并将安全标记名 **$AccessLevel** 设置为 0。

提示 您可以在“条件”QuickScript 中使用 **$InactivityTimeout** 来显示一个窗口，告诉操作员他 / 她将从应用程序注销。

超时功能可独立于**警告**功能来使用。不过，为了让两个系统标记名正常操作，**超时**值必须大于**警告**值。

例如，**警告**在 30 秒闲置后变为真，**超时**在此后 15 秒 （总共 45 秒）闲置后才变为真。

11. 在**计时间隔**框中，输入 InTouch 用于检查内部计时器的速度间隔。

备注 此设置控制应用程序**运行期间**、窗口**显示期间**、条件**为真 / 假期间**、键和触动按纽动作**按下期间**， QuickScript 的执行速度。

在 Windows NT 操作系统中，脚本的执行速度间隔不能超过 10 毫秒，在 Windows 2000 操作系统中不能超过 50 毫秒。

如需有关详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

12. 在**更新时间变量频率**框中，输入希望 WindowViewer 更新基于时间的系统标记名 （如 $Msec, $Second, $Minute 等）的频率 （以毫秒为单位）。

提示 我们建议使用缺省设置 1000 毫秒。不过，您可以输入 0 来阻止更新所有的时间变量。

13. 如果 WindowViewer 中所有的触控对象在选定时发出蜂鸣，请选择**对象接触时蜂鸣**。
14. 如果要加快趋势对象的更新速度，请选择**更新所有趋势 “快速”**。

备注 仅当您完全确认没有对象覆盖运行时趋势对象时，才可选择此选项。如果在有对象覆盖当前趋势对象时显著此选项，则它将无法正常绘制。

15. 如果希望每次执行 QuickScript 时在 Logger 程序中写入一则消息，请选择**调试脚本**。

提示 如果在 **WindowViewer 属性 / 窗口配置**属性页中选择**调试**菜单选项，则可以在运行时从 WindowViewer 的**特别**菜单打开或关闭此命令。

16. 仅当您使用由 InTouch 3.26 或更早版本开发的国际化应用程序时，才可选择**使用旧的 SendKeys**。**Use old SendKeys** is a legacy option and is not used for FactorySuite.
17. 在**慢速、中速和快速**框，输入要用于闪烁动画链接的速度 （以毫秒为单位）。
18. 在 **I/O 重试间隔**框中，输入 InTouch 尝试重新连接 I/O 服务器之前等待的秒数。如果 InTouch 第一次就成功连接上 I/O 服务器，则 **I/O 重试间隔**框将失效。

19. 如果希望在每次启动 WindowViewer 而您试图与之通讯的服务器没有运行时显示一个对话框，请选择**启动本地服务器**选项。例如：



单击**是**以启动服务器，或单击**否**忽略该消息并关闭对话框。

20. 单击**确定**以保存属性设置并关闭对话框。

备注 在修改上述任意参数之后，您必须重新启动 WindowViewer 以便使所作的更改生效。

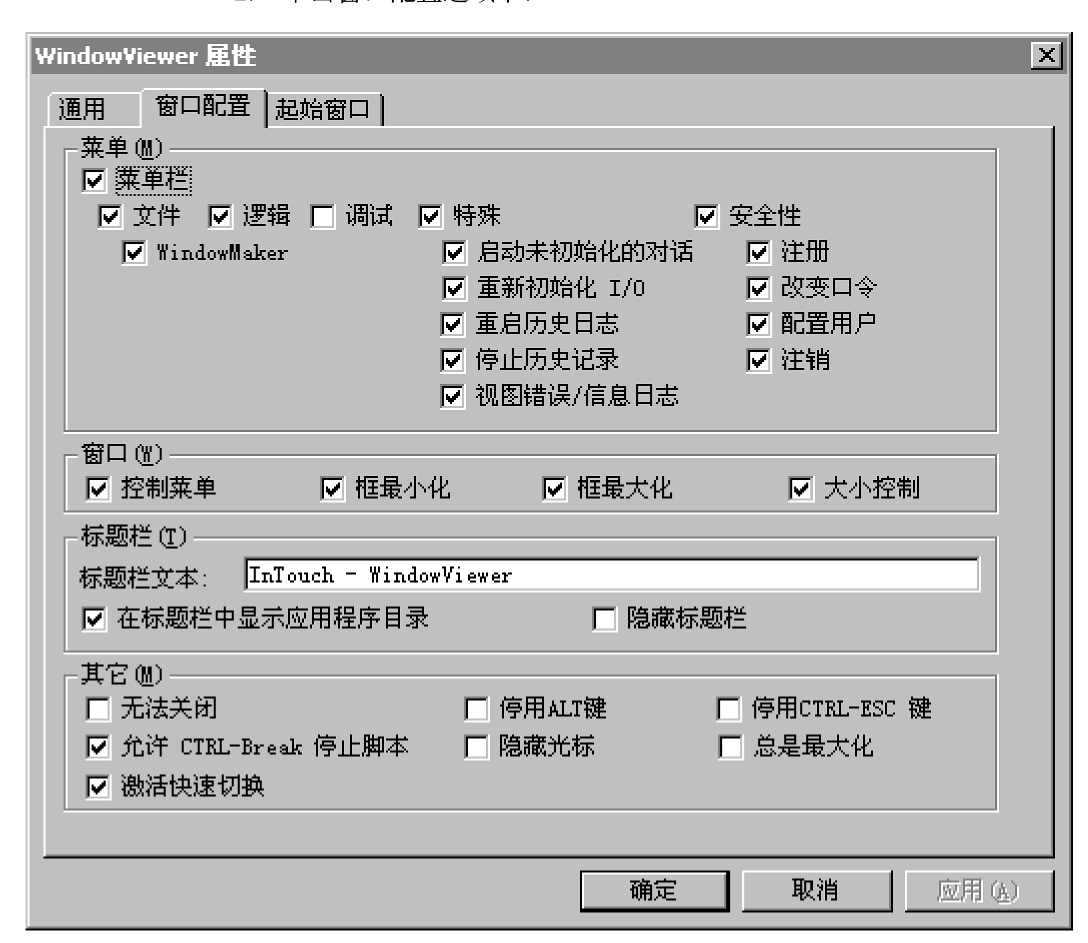
设置 **WindowViewer** 的窗口配置属性

要配置 WindowViewer 程序窗口：

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**，或者在 “应用程序浏览器”中的**配置**下，双击 **WindowViewer**。此时会出现 **WindowViewer 属性**对话框。

提示 在 “应用程序浏览器”中，您也可以右击 **WindowViewer**，然后单击**打开**。

2. 单击**窗口配置**选项卡：



提示 如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

3. 在**菜单**组中，选择希望在运行时显示的菜单和命令。

提示 缺省条件下，在运行 WindowViewer 时会显示菜单栏。清除**菜单栏**选项以防显示菜单栏。

4. 如果想要阻止操作员切换到 WindowMaker 程序，请清除 **WindowMaker** 命令。（选择此选项不影响快速切换到 WindowMaker）。

5. 如果想要防止操作员在运行时启动或停止所有 QuickScripts 的执行，请清除**逻辑**菜单。

备注 您可以使用系统标记名 **$LogicRunning** 来允许操作员启动和停止所有的 QuickScript。

如果选择**允许 CTRL-Break 停止脚本**选项（稍后介绍），则无论**逻辑**菜单是否显示，操作员均可停止所有 QuickScript 的执行。

您不能停止当前执行中的异步 QuickFunction。不过，您可以阻止执行新的异步 QuickFunction。

6. 仅当需要"调试"应用程序时，才可选择**调试**菜单。
7. 选择运行时可用的**窗口**控件。

备注 要隐藏应用程序右上角的关闭按钮（**X** 按钮），则必须清除**控制菜单**选项（也称作系统菜单）。

8. 在**标题栏文本**框中，输入要在运行时显示在应用程序标题栏中的标题。例如：

   ABC Company, Paint APP1

备注 如果使用"推广版许可证"，则不能更改标题栏。

如需有关 FactorySuite 授权的详细信息，请参阅 *FactorySuite 系统管理员指南*。

9. 如果要在标题栏中包括应用程序目录的路径，请选择**显示应用程序目录**。例如：

   ABC Company, Paint APP1 - C:\DEMOAPP1

10. 如果要在运行时隐藏应用程序的标题栏，请选择**隐藏标题栏**。
11. 如果要防止操作员关闭 WindowViewer，请选择**无法关闭**。

备注 要隐藏应用程序右上角的关闭 (X) 框，您还必须清除**控制菜单**选项（也称作**系统**菜单）。

12. 如果希望操作员能够在运行时视情形按 CTRL + BREAK 组合键来终止所有 QuickScripts 的执行，请选择**允许 CTRL-Break 停止脚本**。

备注 您不能停止当前直至执行的异步 QuickFunction。不过，您可以阻止执行新的异步 QuickFunction。

13. 如果要禁用 ALT 键并阻止操作员使用 ALT + 快捷键来执行菜单命令，请选择**禁用 ALT 键**。For example, ALT + F4 to exit the application.

备注 要隐藏应用程序右上角的关闭 (X) 框，您还必须清除**控制菜单**选项（也称作**系统**菜单）。

14. 如果因为使用触摸屏而要防止在运行时显示光标，请选择**隐藏光标**。

15. 如果要防止操作员通过访问 Windows 的**开始**菜单来关闭和（或）切换应用程序，请选择**禁用 ESC 键**与**禁用 WIN 键**。
16. 如果希望 WindowViewer 程序总是保持最大化，请选择**总是最大化**。
17. 单击**确定**以保存设置并关闭对话框。

备注 在修改上述任意参数之后，您必须重新启动 WindowViewer 以便使所作的更改生效。

选择 WindowViewer 的起始窗口

要选择 WindowViewer 的缺省启动窗口

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**，或在“应用程序浏览器”中的**配置**下，双击 **WindowViewer**。此时会出现 **WindowViewer 属性**对话框。

提示 在“应用程序浏览器”中，您也可以右击 **WindowViewer**，然后单击**打开**。

2. 单击**起始窗口**选项卡：



3. 选择在 WindowViewer 直接启动时希望自动打开的窗口。

备注 选择起始窗口不会影响到您用快速切换开关来启动 WindowViewer。直接从图标或菜单命令启动 WindowViewer 时，起始窗口会自动打开。

4. 单击**确定**。

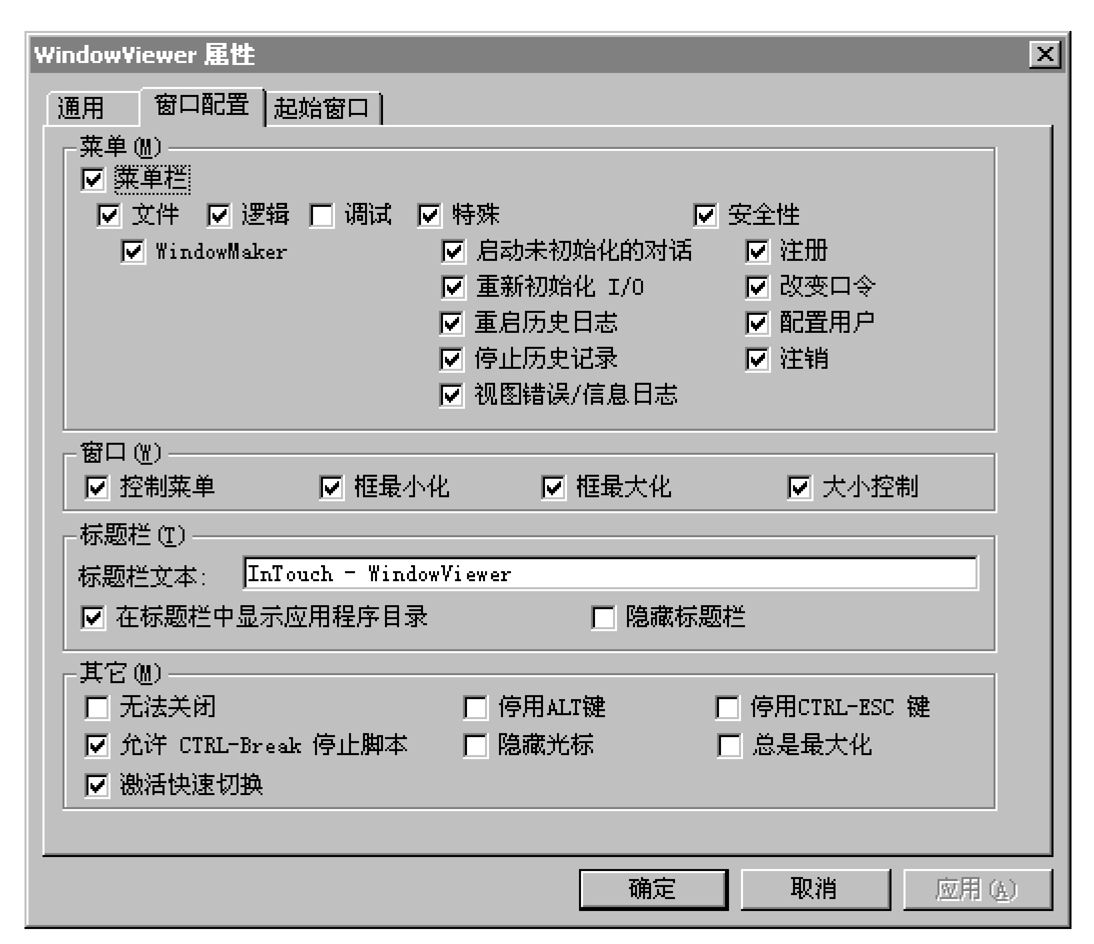
启用键过滤器

应用程序可设计成对于操作系统的功能具有不同级别的访问权限，并在工厂中加以运用。如果您要让各个操作员拥有不同的访问级别，则您可以对其中一些用户启用键过滤器。其他用户 （如系统管理员）则仍可以使用这些键。例如，如果某个没有特殊访问权限的用户使用应用程序，并在控制台上按 Ctrl-ALT-DEL 组合键，则会显示 “Windows 安全”对话框。唯一可用的按钮是**取消**。因此，InTouch 应用程序在运行时，用户无法使用如 “锁定计算机”、“注销”、“关机”、“更改密码”或 “任务管理器”之类的任何选项。用户只能单击**取消**，并返回登录屏幕仍处于显示状态的应用程序。不过，如果管理员使用正确的管理员登录名与密码进行了登录，则这些键不会被禁用。

为了启用键过滤器，您必须确保在应用程序启动时这些键不可操作。

要启用键过滤器

1. 在**特别**菜单上，指向**配置**，然后选择 **WindowViewer**。此时会出现“WindowViewer 属性”对话框。



2. 单击**窗口配置**选项卡。
3. 取消选择**启用快速切换**复选框。
4. 在**其它**字段中，选中**禁用 ALT 键**、**禁用 WIN 键**以及**禁用 ESC 键**复选框，然后单击**应用**。

5. 配置在操作员改变时运行的脚本。该脚本函数按如下方式启用 / 禁用键过滤器：

EnableDisableKeys()

	启用或禁用 Alt、 Escape 及 Windows 键的过滤器	
类别	视图	
语法	`EnableDisableKeys(int AltKey,int EscKey,int WinKey);`	
	参数	**描述**
	AltKey	用以启用或禁用 Alt 键的过滤器的整数， 0 ＝ 禁用、 1 ＝ 启用
	EscKey	用以启用或禁用 Escape 键的过滤器的整数， 0 ＝ 禁用、 1 ＝ 启用
	WinKey	用以启用或禁用 Windows 键的过滤器的整数， 0 ＝ 禁用、 1 ＝ 启用
附注	启用或禁用 Alt、 Escape 及 Windows 键	

示例	`EnableDisableKeys(0,0,0); // 启用所有这三个键`
	`EnableDisableKeys(1,1,1); // 禁用所有这三个键`
	`EnableDisableKeys(0,0,1); // 启用 Alt 和 Escape 键，禁用 Windows 键。`	

将 WindowViewer 作为 NT 服务运行

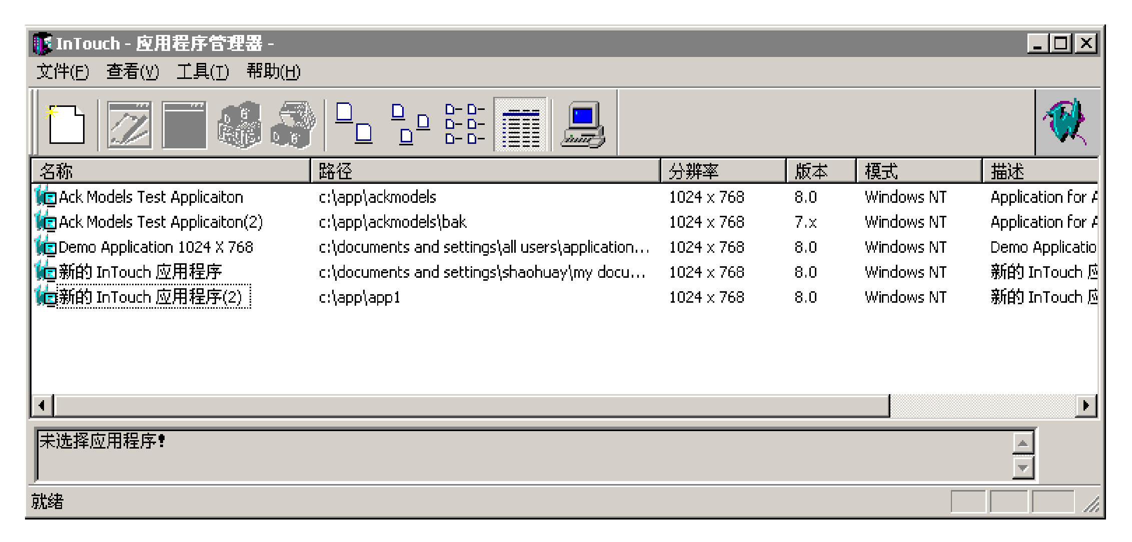
如果使用 InTouch 7.0 （或更改版本），则可以非常轻松地创建客户端 / 服务器配置。您可以将某个节点配置为服务器节点。此服务器节点可以存储 “标记名字典”和历史记录数据、执行 InTouch QuickScript、提供报警机制和 I/O 数据。任何客户端节点随后可以从该服务器节点检索此信息并显示图形。

通过使用 “将 WindowViewer 作为 NT 服务运行”功能，您可以充分利用 “NT 服务”提供的所有功能。例如，在操作员注销后连续作业以及在系统启动期间自动启动，而无须操作员干预。这样，就可以实现在无人工作站中启动 WindowViewer，而又不影响 NT 操作系统的安全性。

如需有关 Windows NT 服务的详细信息，请参阅附录 A “InTouch Windows NT 服务综述”。

要将 WindowViewer 配置成作为 NT 服务启动

1. 启动 InTouch 程序 (intouch.exe)。此时会出现 **InTouch** 应用程序管理器对话框。



2. 单击**节点属性**工具，或者在**工具**菜单上，单击**节点属性**。此时会出现**节点属性**对话框，并显示**应用程序开发**属性页。

提示 要快速访问**节点属性**对话框，请右击显示窗口的空白区，然后单击**节点属性**。

备注 在 “应用程序管理器”窗口中选定某个应用程序时，选择**文件**菜单上的**属性**命令，则可以显示该应用程序的**属性**对话框。



3. 选择**启动 WindowViewer 作为 NT 服务**选项，将 WindowViewer 配置成作为 NT 服务自动运行。

4. 单击**确定**。

备注 如果将 WindowViewer 配置为 NT 服务，并随后直接启动它（使用它的图标或 Windows 开始菜单等），则在大约 15 秒钟的延迟之后，WindowViewer 会显示一个窗口。此延迟是因为 WindowViewer 尝试连接 “NT 服务控制管理器”而造成的。如果未能连接到 “服务控制管理器”，则 WindowViewer 会显示下面的消息框：



如果单击**是**， WindowViewer 将作为应用程序，而不是 NT 服务来启动。如果单击**否**，则取消启动 WindowViewer 的命令。

如果通过关闭**启动 WindowViewer 作为 NT 服务**选项来停止将 WindowViewer 作为 NT 服务运行，则 WindowViewer 服务会被自动卸载。不过，它仍可以作为应用程序运行。

如需有关 Windows NT 服务的详细信息，请参阅附录 A “InTouch Windows NT 服务综述”。

使用 WindowViewer 窗口

InTouch 应用程序通常由一系列窗口组成，这些窗口显示在 WindowMaker 中创建的图形和文本。

本节介绍打开和关闭 InTouch 应用程序中包含的窗口的步骤。

通用对话框特性

当打开或关闭窗口时，所使用的对话框都很相似，它们有许多共同的特性。为避免重复说明如何执行这些操作，本节向您介绍这些对话框的公共特性。

单击**文件**菜单上的**打开窗口**或**关闭窗口**命令时，缺省条件下，所选命令相应的对话框将出现在“列表视图”中。这意味着适合所选命令的所有窗口的名称会出现在一个连续的列表中。例如：



备注 窗口名的数目超过缺省的列表空间时，会出现一个水平滚动条。

单击**详细**可将“列表视图”改为详细视图。选择详细视图时，屏幕将以多列格式显示各个窗口及其详细资料。显示的详细资料包括应用程序开发人员在创建窗口时输入的有关该窗口的任何注释、窗口*的*类型，以及最后一次修改它的日期与时间。例如：



备注 在详细视图中，您可以选择任何一个未打开的窗口，方法是单击它所在行的任何部分，而不只是复选框。（此时整行将辉亮显示）。再次单击选定的窗口，可以取消选择。

窗口名的数目超过缺省的列表空间时，会出现一个垂直滚动条。

要按详细资料类型给列表排序，请单击该详细资料的列标题。详细视图的排序顺序如下：

- **名称** - 按字母顺序排列
- **注释** - 按字母顺序排列
- **类型** - 覆盖、替换、然后才是弹出
- **最近修改日期** － 从上到下按最早到最近的日期 / 时间顺序排列

提示 每次单击列标题时，列表的排列顺序均会从升序变为降序，或正好相反。例如，如果列表当前以升序排列，在您单击某个列标题时，该列表将按降序重新排序选定的列。

要让列表返回缺省显示状态，请单击列标题最左端的小方框。

要调整列大小，请将光标放到每个详细资料列标题的垂直分隔线上。光标变成 "I" 形时，请单击并拖曳该标题至所需宽度。

提示 要快速自动调整列大小，请双击该列*的*右垂直分隔线。

要打开选定的窗口，请单击**确定**。

要取消选择并关闭对话框，请单击**取消**。

要让对话框返回 “列表视图”，请单击**列表**。

要选择列出的所有窗口，请单击**全选**。

要清除选定的所有窗口，请单击**全部清除**。

打开窗口

要打开窗口

1. 在**文件**菜单上，单击**打开窗口**。此时会出现**要显示的窗口**对话框。
2. 单击要打开的窗口名旁边的复选框。

   提示 缺省条件下，当前打开的所有窗口均已选定。

3. 单击 “确定”关闭对话框并打开选定的窗口。

   备注 如果选定了 “替换”型窗口，则会关闭与之相交的所有窗口。

如需有关窗口类型的详细信息，请参阅联机 *InTouch 用户指南*。

关闭窗口

要关闭打开的窗口

1. 在**文件**菜单上，单击**关闭窗口**。此时会出现**要隐藏的窗口**对话框。
2. 单击要关闭的窗口名旁边的复选框。
3. 单击**确定**以关闭对话框并关闭选定的窗口。

转换到 WindowMaker

要从 WindowViewer 程序转换到 WindowMaker 程序

1. 在**文件**菜单上，单击 WindowMaker。此时会出现 “要编辑的窗口” 对话框。

   提示 要快速转换到 WindowMaker，请单击菜单栏右上角的**开发**快速切换开关 （或使用快捷键 ALT + !）。使用快速切换开关进行转换时，**要编辑的窗口**对话框不会出现在 WindowViewer 中。转换到 WindowMaker 时，在 WindowViewer 中打开的窗口仍将处于打开状态。

   备注 仅在开发期间应用程序开发人员已将应用程序配置为使用快速切换开关的情况下，才可使用快速切换开关。

2. 转换到 WindowMaker 时，单击要打开的窗口的名称旁边的复选框。
3. 单击**确定**以关闭对话框并转换到 WindowMaker。

备注 如果在开发期间配置 WindowViewer 的属性时，应用程序开发人员选定了**关闭 WindowViewer** 选项，则转换到 WindowMaker 时，WindowViewer 会自动关闭。

执行 InTouch QuickScript

缺省条件下，初次启动 WindowViewer 时，将会执行所有脚本的逻辑。

要停止执行所有 QuickScript

在**逻辑**菜单上，单击**停止逻辑**。此时会出现**要编辑的窗口**对话框。

备注 在开发期间配置 WindowViewer 时，如果应用程序开发人员选定**允许 CTRL-Break 停止脚本**选项，则无论**逻辑**菜单是否显示，均无法停止执行 QuickScript。

此外，**停止逻辑**命令不会停止任何目前正在执行的异步 QuickFunction，但它可以阻止执行任何新的异步 QuickFunction。

如需有关上述各项的详细信息，请参阅联机 *InTouch 用户指南*。

初始化 I/O 对话

在 WindowViewer 启动时，它会自动处理*初始化*请求来启动所有的 I/O 对话。如果 “I/O 服务器”程序未对 WindowViewer 的*初始化*请求作出响应，则您可以强制 WindowViewer 再次尝试建立 I/O 对话。

要启动所有尚未初始化的 I/O 对话

在**特别**菜单上，单击**启动未初始化的对话**。

提示 执行此命令不会影响现有的对话。

要重新启动所有的 I/O 对话

在**特别**菜单上，单击**重新初始化 I/O**。

提示 此命令将关闭现有的所有 I/O 对话，并重新启动设置 I/O 对话的整个过程。此命令会影响所有的 I/O 数据点。

第 3 章

使用 **InTouch** 安全性

InTouch 既给您提供了选择传统型基于 InTouch 的安全性选项，也给您提供了选择基于操作系统或 ArchestrA 的安全性选项。所有的 InTouch 安全方法都可以在应用程序级配置，这表示，您可以在同一台计算机上配置两个使用不同安全设置的应用程序。

所有这三种安全方法均与应用程序的 “网络应用程序开发” (NAD) 分布兼容。同以前版本的 InTouch 一样，基于 InTouch 的安全性可以结合 NAD 使用。如需有关 NAD 的详细信息，请参阅网络应用程序开发 (NAD)。不管是否使用 NAD，基于 ArchestrA 的安全性均采用集中化管理。

如果身份验证模式基于操作系统，则用户名将会是 Windows 的 “域名 / 用户名”对。如果采用基于 ArchestrA 的模式，则与安全性有关的活动将会在 “集成开发环境”（Intergrated Development Enviroment，简称 IDE）外部配置。如需有关 IDE 的详细信息，请参阅 《Wonderware® ArchestrA™ 集成开发环境 (IDE) 指南》。

目录

使用基于 **InTouch** 的安全性

给应用程序应用安全性属可选项。InTouch 应用程序的缺省安全设置是 “无”。不过，通过给应用程序应用安全性，您可以通过链接特定的函数与内部标记名，而控制允许操作员执行的功能。此外，您在应用程序中建立安全性时，可以创建审查跟踪线索，将操作员与其登录到系统后发生的所有报警 / 事件联系起来。

安全性的基本概念是操作员 “登录”到应用程序，然后输入其用户名与口令。因此，您必须给每个操作员配置用户名、口令及访问级别。

Microsoft 操作系统安全性与 InTouch 安全性之间没有任何联系。

创建新的应用程序时，缺省条件下，用户名会被设置为 "Administor"，其访问级别是 9999 （此级别允许访问所有的安全性命令）。将新用户名添加到安全性列表并重新启动 WindowMaker 或 WindowViewer 之后，缺省用户名会被自动重设为 “None”而其访问级别会被设置为 "0" （此级别不允许访问 WindowMaker 和 WindowViewer 中的**配置用户**命令）。不过， Administrator 帐户与口令仍在，并且仍可以使用。

在操作员登录到应用程序之后，访问任何受保护的功能时，系统均会根据给链接到该功能的内部安全性标记名指定的值去验证操作员的口令和访问级别，以确定是否授予访问权。

例如，通过指定登录操作员的 “访问级别”必须大于 2000，您就可以控制对某个窗口的访问，或某个对象的可见性等。

通过执行 WindowViewer **特别**菜单 （如果**特别**菜单有显示）中**安全性**下的**登录**菜单命令，操作员可以登录到应用程序，此外，您也可以创建一个自定义登录窗口，在该窗口中放上链接到内部安全性标记名的触控输入对象。

用于建立应用程序安全性的命令位于 WindowMaker 和 WindowViewer **特别**菜单上的**安全性**下。安全性命令用于登录到应用程序或从应用程序注销、更改口令，以及配置有效用户名、口令和访问级别的列表。

使用安全内部标记名

在应用程序中实施安全性之后，您便可以在按钮、动画链接表达式或 QuickScript 等对象上使用多个内部安全性标记名，控制是否允许登录的操作员执行特定的功能：

标记名	类型	有效值	访问权限
$AccessLevel	系统整型	0-9999	只读
$Operator	系统消息型	最大 16 个字符	只读
$ChangePassword	系统离散型	1 or 0	读写
$ConfigureUsers	系统离散型	1 或 0	读写

标记名	类型	有效值	访问权限
$InactivityTimeout	系统离散型	1 或 0	只读
$InactivityWarning	系统离散型	1 或 0	只读
$OperatorEntered	系统消息型	最大 16 个字符	读写
$PasswordEntered	系统消息型	最大 16 个字符	读写

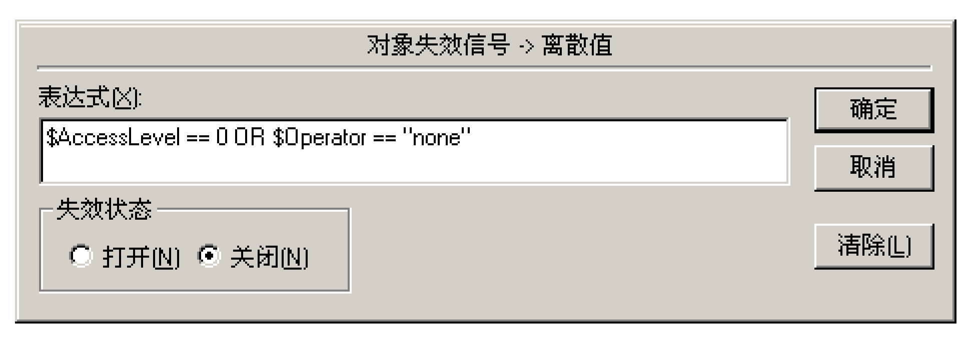
例如，要根据登录用户的访问级别来显示对象，您可以在可见性动画链接表达式中使用下面的语句：

```
$AccessLevel >= 2000;
```

或者是，使用由 **IF** 语句进行限定的 **QuickScript**：

```
IF $Operator == "DayShift" THEN
   Show "Control Panel Window";
   { 以及仅给 “日班”操作员执行的其它代码行 }
ENDIF;
```

通过使用**失效**动画链接，您还可以基于内部安全性标记名的值控制对象的触动功能。例如：



通过使用此表达式，如果无人登录，则该对象或按钮会被禁用，以防意外操作。

如需有关内部安全性标记名的详细信息，请参阅您的联机 *InTouch 参考指南*。

配置用户与安全级别

要配置应用程序操作员的安全级别

1. 在**特别**菜单上，指向**安全性**，然后单击**配置用户**。此时会出现**配置用户**对话框。



提示 如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

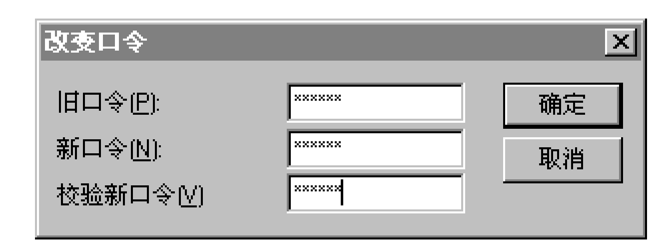
2. 在**用户名**框中，输入要指定给操作员的名称。
3. 在**口令**框中，输入口令 （最长 16 个字符）。
4. 在**访问级别**框中，输入一个值 （最低 = 0，最高 = 9999）。
5. 单击**添加**将该用户名添加到安全性列表。
6. 要**修改**现有的用户名，请从**用户名**列表选择所需的名称。输入更改，然后单击**更新**以接受更改。
7. 要**删除**某个用户名，请在列表中选定它，然后单击**删除**。

None 和 **Administrator** 名称是保留的，只有 Administrator 的口令才能更改。为应用程序配置了用户名之后，您应该更改 Administrator 用户名的口令，因为它很可能已泄露给系统中的大多数用户。Administrator 的缺省访问级别 (9999) 是最高的级别，可以访问包括 “配置用户”菜单命令在内的所有内容。

更改安全登录口令

要更改操作员的口令

1. 在**特别**菜单上，指向**安全性**，然后单击**改变口令**。此时会出现**改变口令**对话框。



   如果右击任意对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 在**旧口令**字段中，输入旧口令。
3. 在**新口令**字段中，输入新口令 （最长 16 个字 符）。
4. 在**验证口令**字段中，再输入一次新口令。
5. 单击**确定**。

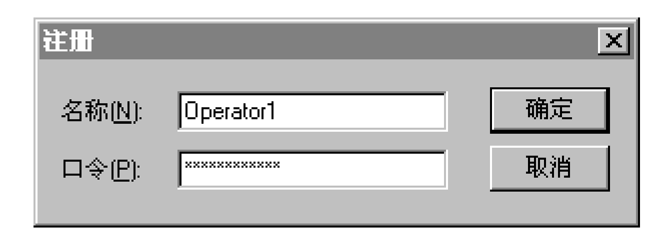
   为防止其它任何人看到操作员输入的口令，所输入信息在屏幕上均显示为星号。

提示 如果您不打算在 WindowViewer 中显示**特别**菜单，则可以创建一个离散型按钮，并将它链接到 **$ChangePassword** 内部标记名。这样做可以将 **$ChangePassword** 标记名设置为 1，以便显示**改变口令**对话框。显示该对话框之后，操作员就可以更改自己的口令了。

登录到基于 **InTouch** 安全性的应用程序

要登录到应用程序

1. 在**特别**菜单上，指向**安全性**，然后单击**登录**。此时会出现**登录**对话框。



2. 在**名称**框中，输入您的用户名。
3. 在**口令**框中，输入口令。

4. 单击**确定**。

   如果输入的信息不正确或无效，则会出现一消息框，指出登录失败。

   如果登录成功，则 **$AccessLevel** 内部标记名会被设置为其预设值（在安全性用户列表中配置）。

备注 另请参阅 PostLogonDialog()。

注销基于 **InTouch** 安全性的应用程序

要注销应用程序

- 在**特别**菜单上，指向**安全性**，然后单击**注销**。

执行此命令时，“用户名”会被重置为 “无”，访问级别会被重置为 "0"。

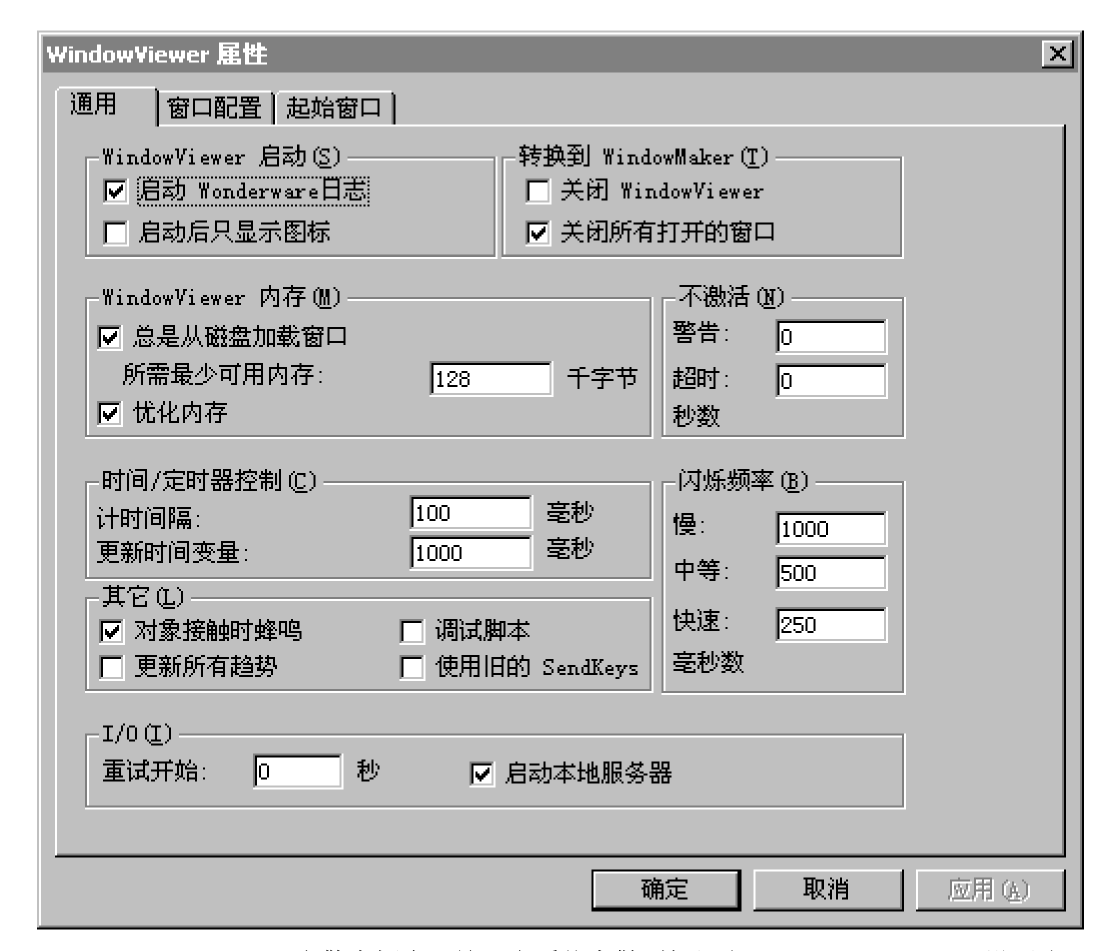
您还可以将应用程序配置成在操作员停止活动超过指定的时间量之后自动将其注销。

自动注销系统

通过使用警告与超时设置，您可以将应用程序配置成在操作员未执行任何操作超过指定的时间之后自动注销该操作员。

要配置闲置

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**，或者在 “应用程序浏览器”中的**配置**下，双击 **WindowViewer**。此时会出现 **WindowViewer 属性**对话框，并显示**通用**属性页。在 “应用程序浏览器”中，您也可以右击 **WindowViewer**，然后单击**打开**。



2. 在**警告**框中，输入在系统离散型标记名 **$InactivityWarning** 设置为 1(True) 之前无操作员活动 （鼠标或击键）的时间 （以秒计）。在**不活动警告**被设置为零时，便不会有任何闲置警告发出。

提示 您可以在 “条件” QuickScript 中使用 **$InactivityWarning** 来显示一个窗口，以警告操作员即将从系统注销。如果操作员在指定的超时范围内，单击鼠标、击键或者使用其它定向设备执行任一动作，则它将不会注销。 **$InactivityWarning** 与计时器均会重置。

3. 在**超时**框中，输入系统离散型标记名 **$InactivityTimeout** 设置为 1(True) 之前无操作员活动 （鼠标或击键等）的时间 （以秒计）。在 **$InactivityTimeout** 为真时，系统会将登录的操作员用户名视为保留名 "None"，并将安全标记名 **$AccessLevel** 设置为 0。

提示 您可以在 “条件” QuickScript 中使用 **$InactivityTimeout** 来显示一个窗口，告诉操作员他 / 她将从应用程序注销。

超时功能可独立于**警告**功能使用。不过，为了让两个系统标记名正常操作，**超时**值必须大于**警告**值。

例如，将 $InactivityWarning 设置为 30 并将 $InactivityTimeout 设置为 45。在 $InactivityWarning 变量被设置为 1 的 15 秒之后，操作员将会注销。

使用基于操作系统的安全性

在基于操作系统的身份验证架构中，可以从与 Windows “网络域” \ “工作组”关联的用户列表中选择用户名。每个用户名均被指定一个访问级别，由它确定用户就给定活动所拥有的权限。因为操作系统在内部管理口令，因此 InTouch 不存储口令。

基于操作系统的安全性使用 InTouch 脚本函数 AddPermission 来维护用户及其相应访问级别的列表。此列表在执行 Addpermission() 调用创建，它会被写入磁盘。包含用户身份验证详细资料不会被复制到 NAD 客户端机器。

设置基于操作系统的安全性

基于操作系统的安全性可以从 WindowMaker 的 “选择安全类型”菜单中选择。这通常会在创建新应用程序时完成。通常，您会在创建新的 InTouch 应用程序时选择安全性设置。

要设置基于操作系统的安全性

1. 在 WindowMaker 中打开一个窗口。
2. 在**特别**菜单上，指向**安全性**，然后指向**选择安全类型**并选择**操作系统**。

使用安全性系统标记名

基于操作系统的安全性有三个新的安全系统标记名，它们分别是 $OperatorDomain、 $OperatorDomainEntered 以及 $OperatorName。下表介绍这些新系统标记的功能，并提供一些示例：

$OperatorDomain

类别	**安全性**
用法	如果选择了基于操作系统的安全性，并且操作员已成功登录，则 $OperatorDomain 标记将包含登录时指定的域名或机器名。如果用户在登录时选定了 Archestra 安全性，则 $OperatorDomain 将包含 "ArchestrA"。如果选定了 InTouch 安全性，则 $OperatorDomain 标记包含字符串 "InTouch"。如果选定 "无"，则它是一个空字符串 ""。
附注	N/A
数据类型	字符串
示例	`$Operator = "john";` `$OperatorDomain="CORPORATE_HQ";`
另请参阅	**$Operator**

$OperatorDomainEntered

类别	**安全性**
用法	只要 $PasswordEntered 标记发生变化，便会尝试从内部登录，而不显示任何图形用户界面。登录尝试将 $*Entered 标记用作输入用户名，将字符串值 $OperatorDomainEntered 用作域名 （仅在当前模式是基于操作系统的安全性时使用）。如果安全模式并非基于操作系统，则会忽略此标记。
附注	N/A
数据类型	字符串

示例	`$OperatorEntered="john";`
	`$OperatorDomainEntered="Corporate_hq"`
	`$PasswordEntered="password";`

另请参阅	**$Operator**

$OperatorName

类别	安全性
用法	如果采用基于操作系统或 ArchestrA 的身份验证方式，并且有人已经登录且尚未注销，则 $OperatorName 标记将包含操作员的全限定名称。否则，该标记将包含登录用户的名称 （与 $Operator 标记的内容相同）。
附注	N/A
数据类型	字符串
示例	`$Operator = "john";$OperatorName = "John Smith";`
另请参阅	**$Operator**

$VerifiedUserName

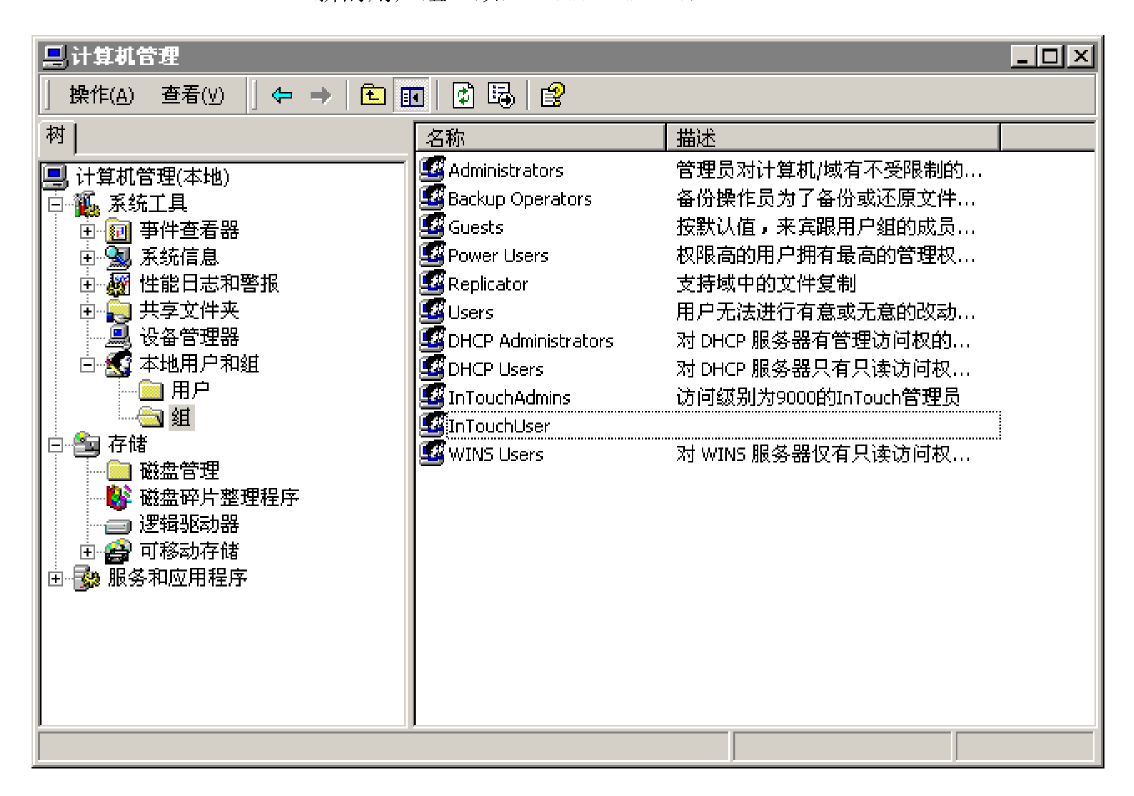
	如果调用 InvisibleVerifyCredentials() 成功，且安全模式被设置为基于操作系统或基于 ArchestrA AppServer 的安全性，则包含经验证的用户的全名。如果调用失败，则上述系统标记会被设置为空。
类别	安全性
用法	**$VerifiedUserName**
附注	只要上述系统标记发生改变 （表示只要调用了 InvisibleVerifyCredentials），便会产生一个事件，并且如果调用成功，则 "Value" 列将包含经验证的用户的全名。如果调用失败，则 "Value" 列将包含空值。 Name 列包含的值会是 "$VerifiedUserFullName"。
数据类型	字符串
有效值	**用户的全名**
示例	Tag = InvisibleVerifyCrdentials("john","password", "Plant_Floor"). 如果调用成功，则 $VerifiedUserName 会被设置为 "John Smith"，并且会生成一个“操作员事件”。此时，name 列被指定为 $VerifiedUserName，并且 value 列被设置为 "John Smith"。如果上述调用不成功，则在该事件中 $VerifiedUserName 和 value 列均会被设置为 ""。 每次调用上述脚本函数时， $VerifiedUserName 均会被设置为相应用户的全名，或是被设置为空值。
另请参阅	InvisibleVerifyCredentials(); $OperatorName, $Operator

设置用户组

基于操作系统的安全性使用授权的 Windows 用户组的列表。用户将在本地计算机或域服务器上创建用户组。管理员必须通过将 Windows 用户添加到指定的组而将他们同用户组相关联。在 WindowMaker 中，应用程序开发人员必须使用 AddPermission() 函数为每个组指定所需的访问级别来设置用户组列表。AddPermission() 通常会在应用程序启动时调用，这样在用户准备登录时 View 就可以识别所有经授权的用户组。您必须作为本地管理员登录，或是具有本地管理员权限，如此才能设置并管理本地计算机上的组。

要在本地计算机上设置管理员组并添加用户

1. 在**开始**菜单上，单击**设置**，然后单击**控制面板**。
2. 双击**管理工具**图标，然后双击**计算机管理**图标。此时会出现 “计算机管理” 窗口。
3. 在操作系统的 “计算机管理” 窗口中，为 InTouch 管理员创建一个新的用户组 （如 InTouchAdmins）。



4. 从**操作**菜单，单击**新建组**，或右击 “计算机管理”屏幕的右侧窗格，然后单击快捷菜单上的**新建组**。此时会出现**新建组**对话框。



5. 在**组名**框中，给您的组输入一个名称，然后如果需要请在**描述**框中输入描述。

6. 单击**添加**。此时会出现**选择用户或组**窗口。

7. 单击要添加的成员的名称，然后单击**添加**。

提示 要添加多个成员，请按住 CTRL 键，单击附加的成员名，然后再单击**添加**。

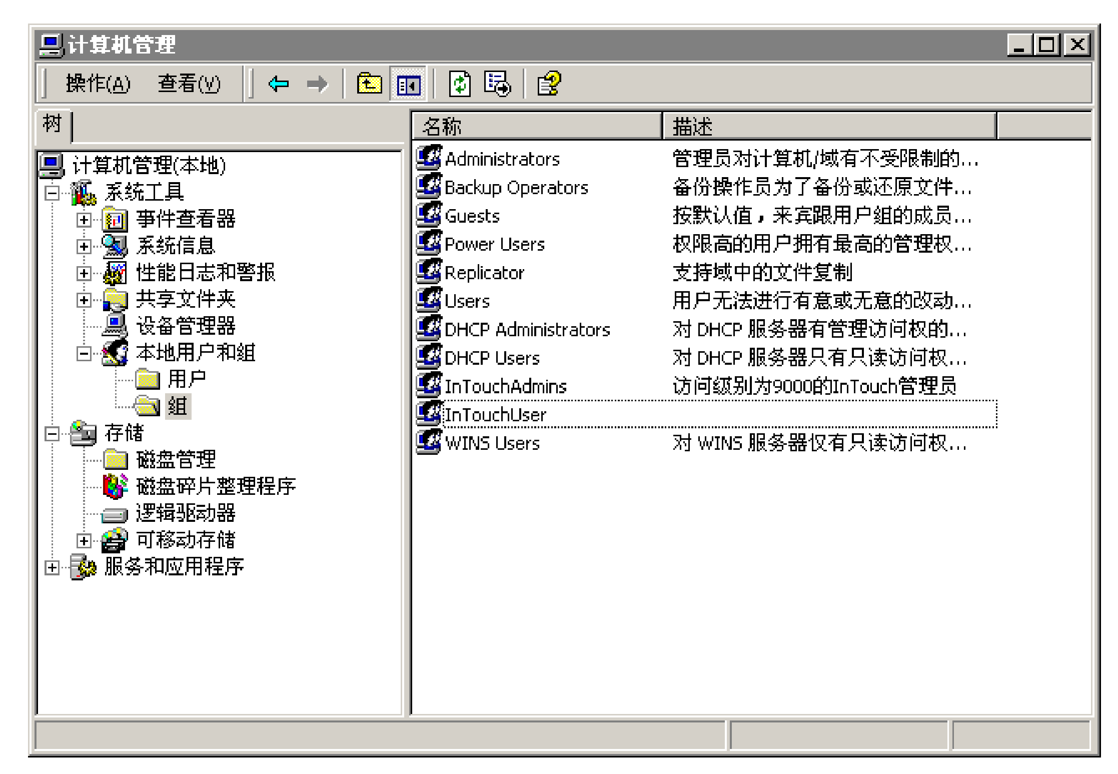
---

8. 单击**确定**。

要在本地计算机上设置用户组并添加用户

1. 在**开始**菜单上，单击**设置**，然后单击**控制面板**。
2. 双击**管理工具**图标，然后双击**计算机管理**图标。此时会出现 “计算机管理” 窗口。

3. 在操作系统的**计算机管理**窗口中，为 InTouch 用户创建一个新的用户组 （如 InTouchUsers）。



4. 从**操作**菜单，单击**新建组**，或右击 “计算机管理” 窗口的右侧窗格，然后单击快捷菜单上的**新建组**。此时会出现**新建组**对话框。

5. 在**组名**框中，给您的组输入一个名称，然后如果需要请在 “描述” 框中输入描述。

6. 单击**添加**。此时会出现**选择用户或组**窗口。

7. 单击要添加的成员的名称，然后单击**添加**。

提示 要添加多个成员，请按住 CTRL 键，单击附加的成员名，然后再单击**添加**。

---

8. 单击**确定**。

要将用户添加到现有的组

1. 双击组名，以查看组属性对话框。此时会出现该组的**属性**对话框。



2. 单击**添加**以添加用户。此时会出现**选择用户或组**窗口。
3. 单击要添加的成员的名称，然后单击**添加**。

提示 要添加多个成员，请按住 CTRL 键，单击附加的成员名，然后再单击**添加**。

---

4. 单击**确定**。

如需有关创建用户组的详细信息，请参阅 Windows 操作系统文档。

您将 InTouch 应用程序配置为利用操作系统的 “身份验证”与 InTouch 的内部 “身份验证”之后，**改变口令**、**登录**、**配置用户**及**注销**等选项在**特别** ... **安全性**菜单上便不再可用。

在 WindowMaker 中设置组的访问级别

给 InTouch 设置基于操作系统的安全性的最后一个步骤是在 WindowMaker 中设置组的访问级别。在 WindowMaker 中，AddPermission() 用于使用所需的访问级别设置组列表。

要设置组的访问级别

1. 启动 WindowMaker。
2. 在**特别**菜单上，指向**脚本**，然后单击**应用程序脚本**。
3. 在**条件类型**列表中，单击**启动时**。
4. 使用 AddPermission()，输入组名与相应的访问级别。AddPermission() 所需的缺省参数是操作系统或 “域”、“组” 以及 “访问级别”。



5. 单击**确定**。

InTouch 操作系统安全性函数

基于操作系统的身份验证架构从操作系统继承了操作系统一些帐户策略的实行方式，而其它一些策略则是在 InTouch 中实行。口令策略 （如最大与最小口令寿命，以及最小口令长度）则是由操作系统实行的。在安装期间使用的用户名会作为操作系统的一部分发挥作用。Windows 域必须使用所需的帐户策略进行设置，以实行这些标准。InTouch 实行不活动超时策略。

登录到基于操作系统安全性的应用程序

用户登录到 InTouch 应用程序时，会出现一个对话框，要求输入用户名、口令以及域或本地计算机名。 “域 / 用户名”组合会被传递到操作系统以验证该帐户。此时，不管是否启用操作系统缓冲区，均会尝试登录一次。如果不使用缓冲区 （由于网路中断）用户无法登录，但先前使用缓冲区时用户曾通过身份验证，则会从本地的 InTouch 缓冲区获取该用户的全名与访问级别。如果成功通过所有检查，则该用户会被视为已登录到 InTouch，相关的数据结构 （如 $Operator）也会被更新。否则，会显示一则相应的错误消息。

备注 如果该操作员先前从未成功登录，且域不可用，则该次登录会失败。

使用基于 ArchestrA 的安全性

将节点配置为使用 ArchestrA 安全性时， InTouch 会从 AppServer 调用方法与对话框，以配置用户或进行登录 / 注销。这些用户是在 AppServer Galaxy 节点上配置的。如需有关详细信息，请参阅 AppServer 文档。

关于 ArchestrA 身份验证与授权

按照其设计理念， ArchestrA Security 系统可供系统管理员轻松定义系统用户，并指定允许他们执行的操作。这些安全权限是使用这些用户界面工具基于用户可以执行的操作进行定义的。这个基本的解决方案由下面几个步骤组成：

1. 定义安全模型。
2. 根据安全模型组织整理要保护的自动化对象。
3. 根据安全模型定义用户。

系统管理员通过创建相应的用户配置文件来定义系统用户。随后，系统管理员通过从安全模型中预先定义的用户角色列表中进行选择来定义这些用户的角色 （一个用户可以有多种角色）。

InTouchView 用户通常会通过基于口令的方式进行身份验证。此外，还会提供一个安装期间使用的选项，以使用 Windows 登录进行身份验证。当然，这便要求在 Windows 操作系统中自定义该用户。

此外，还有一个以 ActiveX 控件的形式提供的用户身份验证实用程序，可供第三方给 ArchestrA 开发客户端应用程序以支持 ArchestrA 安全机制。

设置基于 ArchestrA 的安全性

在管理员已经为 InTouch 或 InTouchView 应用程序的用户定义用户配置文件之后，管理员便可设置 ArchestrA

要设置基于 ArchestrA 的安全性

1. 在 WindowMaker 中打开一个窗口。
2. 在**特别**菜单上，指向**安全性**，然后指向**选择安全类型**并选择 **ArchestrA**。

备注 您将 InTouch 应用程序配置为利用 ArchestrA 的身份验证与授权方式之后，**改变口令**、**登录**、**配置用户**及**注销**等选项在**特别 ... 安全性**菜单上便不再可用。

InTouch ArchestrA 安全性函数

ArchestrA 安全系统是一个全局函数，适用于 Galaxy 数据库中所有的对象。它是用户同 Galaxy 中的对象与函数之间的一个基于关系的系统。此系统基于安全角色 （配置、系统管理、运行时权限）与安全组，以确定特定安全角色的对象级运行时权限。安全系统的配置是在 GalaxyObject 的编辑器中完成的，并通过它自己的编辑器应用于每个对象。

登录到基于 ArchestrA 安全性的应用程序

用户通常会通过输入有效的用户名与口令来登录与注销基于 ArchestrA 安全性的 InTouch 应用程序。

如果您的系统是使用开放式安全性配置的，则会使用缺省用户的登录凭证，而不是提示您去登录。下列操作过程假设您的系统已配置为采用身份验证模式的安全性。

要进行登录，请按照下列步骤进行操作：

1. 启动基于 ArchestrA 的 InTouch 应用程序。此时会显示一个登录对话框。

2. 输入有效的用户名与口令。如果系统无法对您进行身份验证，则会提示您再次登录。

在系统验证完您的登录数据之后，对将来所有操作的访问权限是基于 “安全模型”中关联的角色 / 权限而授予的。

创建自定义安全登录窗口

如果 WindowViewer 中没有显示**特别**菜单，则您可以创建一个自定义登录窗口让操作员登录到应用程序。

要创建自定义登录窗口

- 将 **$OperatorEntered**、 **$PasswordEntered** 及 **$OperatorDomainEntered** 系统标记名链接到用户输入对象，或者在 QuickScript 中使用它们，以设置 “用户名”、“口令” 及 “域名” （这些是特意仅为写入操作提供的内部消息型标记名）。仅当安全模式基于操作系统时，才要求使用 **$OperatorDomainEntered**。否则，此标记会被忽略。如果安全模式是基于操作系统的，且 **$OperatorDomainEntered** 为空，则它会被视为指向本地机器。

例如：

将 “用户名” 字符串设置为 `-> $OperatorEntered`

将 “用户域名” 字符串设置为 `-> $OperatorDomainEntered`

将 “用户口令” 字符串设置为 `-> $PasswordEntered`

与 $OperatorEntered 和 $PasswordEntered 不同， $OperatorDomainEntered 值发生改变时不会触发登录事件。

如果这些项目有效，则 **$AccessLevel** 和 **$Operator** 内部标记名将会被设置为它们的预设值 （在安全性用户列表中配置）。

此外，如果 WindowViewer 中没有显示**特别**菜单，则您可以链接**用户输入 - 离散**按钮与 **$ChangePassword** 标记名，以显示**改变口令**对话框供操作员更改其口令。在操作员单击按钮时， **$ChangePassword** 标记名的值会被设置为 1，并且会出现**改变口令**对话框。操作员关闭该对话框时，系统会将该值重置为 0 （这是一个仅用于写入操作的系统离散型标记名）。

您还可以链接**用户输入 - 离散**按钮与 **$ConfigureUsers** 标记名，以便其访问级别大于或等于 9000 的授权操作员可以访问**配置用户**多个可来编辑安全用户名列表。在操作员单击按钮时， **$ConfigureUsers** 标记名的值会被设置为 1，并且会出现**配置用户**对话框。操作员关闭该对话框时，系统会将该值重置为 0 （这是一个仅用于写入操作的系统离散型标记名）。

安全与报警

将 InTouch 报警供应器配置为使用基于操作系统或 ArchestrA 的身份验证方式，并且发生报警时，报警显示对象将在 “操作员”列包含操作员的全名，只要该操作员已登录。例如，如果某个用户是在 PLANT_FLOOR 域注册的，并且其 UserID 为 JohnS，全名为 John Smith，则 “操作员”列将包含 John Smith 字样。如果随后确认了该报警，并且执行确认的节点被设置为使用操作系统或 ArchestrA 安全性，则 “操作员”列会被更新为确认操作员的全名。否则，报警显示对象会显示与 $Operator 标记中的内容有关的计算机名

报警记录中的全名扩展

InTouch 安全性可以在确认报警时提供操作员的全名。这在属于报警检测的记录上也是可能的。在大多数机构中，“登录 ID”并非某个人的全名，而是缩写或者是角色类别。

在供应器与接收器 InTouch 节点上选定操作系统身份验证方式时：

- 在生成报警并执行确认时，报警显示对象会显示全名。
- 在生成报警并执行确认时，报警打印对象会打印全名。
- Alarm DB Logger 会随每个报警记录在 operator 与 AckOperator 字段中记录域名、登录用户 ID 以及用户的全名。这样，即便某个机构中有两个员工的全名完全相同，也可以作出唯一的判断。
- 在网路上发送报警与确认数据包时，域名与登录用户 ID 会连接到现有的操作员用户名字段中。

InTouch 安全性脚本函数

InTouch 具备了一些同安全性有关的新函数。下文将介绍这些新函数。

InvisibleVerifyCredentials()

	选择此项时，可以在不将用户登录到 InTouch 的情况下验证给定用户的凭证。	
类别	安全性	
语法	`AnalogTag=InvisibleVerifyCredentials( "UserId", "Password", "Domain" );`	
	参数	描述
	UserId	Windows 操作系统用户帐户名，它属于本地机器、工作组或域的一部分。

附注	如果提供的用户、口令和域的组合有效，则会将同该用户关联的相应访问级别作为整数来返回，在所有其它情况下，均返回 -1。此调用不会改变当前已经登录的用户。“域”字段仅对于基于操作系统的安全模式而言是有效的。如果在使用 ArchestrA 安全模式，并且 ArchestrA 安全性又用到了基于操作系统的安全性，则 “用户 ID”应包含带有域名或计算机名的全限定用户名。
示例	`AnalogTag=InvisibleVerifyCredentials( "john", "Password", "corporate_hq" );`
另请参阅	**PostLogonDialog(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission().**

PostLogonDialog()

	启动 InTouch “登录”对话框，并返回 TRUE。	
类别	安全性	
语法	`DiscreteTag=PostLogonDialog();`	
	参数	描述
	N/A	N/A
附注	启动 InTouch “登录”对话框，并返回 TRUE。	
示例	`DiscreteTag=PostLogonDialog();`	
另请参阅	**InvisibleVerifyCredentials(), AttemptInvisibleLogon(), IsAssignedRole(), QueryGroupMembership(), AddPermission().**	

AttemptInvisibleLogon

	试图使用所提供的凭证登录。	
类别	安全性	
语法	`DiscreteTag=AttemptInvisibleLogon( "UserId", "Password", "Domain" );`	
	参数	描述
	UserId	一个有效的用户帐户名。
	Password	用户的口令。
	Domain	用户所属的本地机器、工作组或域的名称。只有在当前安全类型基于操作系统时，此列才适用。
	DiscreteTag	返回值：如果通过身份验证，则返回 TRUE。否则它返回 FALSE。

附注	试图使用所提供的凭证登录 InTouch （如果安全模式并非基于操作系统，则会忽略 “域”）。如果登录成功，则返回 TRUE，且会相应地更新 $OperatorDomain、 $OperatorName、 $AccessLevel 及 $Operator 等系统标记。如果登录失败，则返回 FALSE，且当前已经登录的用户 （如果存在）仍继续是当前用户。 “域”字段仅对于基于操作系统的安全模式而言是有效的。如果在使用 ArchestrA 安全模式，并且 ArchestrA 安全性又用到了基于操作系统的安全性，则 “用户 ID”应包含带有域名或计算机名的全限定用户名。
示例	`AnalogTag=AttemptInvisibleLogon( "john", "Password", "corporate_hq" ); \\ 安全性基于操作系统，` `AnalogTag=AttemptInvisibleLogon( "john", "Password", "" ); \\ 安全性基于 InTouch 或 ArchestrA。`
另请参阅	**PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), QueryGroupMembership(), AddPermission().**

IsAssignedRole

类别	安全性	
语法	`DiscreteTag=IsAssignedRole( "RoleName" );`	
	参数	描述
	RoleName	与 AppServer 用户关联的角色。
附注	仅在 ArchestrA 安全模式下有效，并且仅适用于当前登录的用户。如果某个用户目前已经登录，且在 Galaxy IDE 中他被指定 RoleName 角色，则会返回 TRUE。在所有其它情况下，均会返回 FALSE。	
示例	`DiscreteTag=IsAssignedRole( "Administrator" );`	
另请参阅	**AttemptInvisibleLogon(), PostLogonDialog(), InvisibleVerifyCredentials(), QueryGroupMembership(), AddPermission().**	

QueryGroupMembership()

类别	安全性	
语法	`DiscreteTag=QueryGroupMembership( "Domain", "Group" );`	
	参数	描述
	Domain	组所在域或本地机器的名称
	Group	组所在域或本地机器的名称。

附注	仅在 AppServer 安全模式下有效，并且仅适用于当前登录的用户。如果某个用户目前已经登录，并且他属于 Group 组，该用户组位于 Domain 上，则会返回 TRUE ；在所有其它情况下，均会返回 FALSE。仅当 ArchestrA 安全性被设置为基于操作系统组的安全性时， QueryGroupMembership 才能在 InTouch 操作系统安全性及 InTouch ArchestrA 安全性下正常工作。
示例	`DiscreteTag=QueryGroupMembership( "corporate_hq", "InTouchAdmins" ); DiscreteTag=QueryGroupMembership( "JohnS01", "InTouchUsers" );`
另请参阅	**BOOL PostLogonDialog(), InvisibleVerifyCredentials(), BOOL IsAssignedRole(), AttemptInvisibleLogon(), AddPermission().**

AddPermission()

	试图访问 Domain 域上的 Account 帐户	
类别	安全性	
语法	`DiscreteTag=AddPermission( "Domain", "Group", AccessLevel);`	
	参数	描述
	Domain	组所在域或本地机器的名称。
	Group	Windows 用户组。
	AccessLevel	与给定的组关联的 InTouch “访问级别”。
附注	仅对于操作系统安全模式有效。试图访问 Domain 域上的 Account 帐户。如果成功，则返回 TRUE，且 AccessLevel 访问级别会被指定给 InTouch 内部记录中的帐户，以便在用户登录时的授权过程中使用。在所有其它情况下，均会返回 FALSE。	
示例	`DiscreteTag=AddPermission( "corporate_hq", "InTouchAdmins", 9000);DiscreteTag=AddPermission( "johns01", "InTouchUsers", 5000);`	
另请参阅	**PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership().**	

ChangePassword()

	显示**改变口令**对话框，供已经登录的操作员更改他 / 她的口令。	
类别	安全性	
语法	`[Result=]ChangePassword();`	
	参数	描述
	[Result]	返回下列整数值之一：
		0 = 已按过 “取消”。
		1 = 已按过 “确定”。

附注	如果使用触摸屏应用程序，则会提供一个使用字母数字键盘的选项。
示例	`Errmsg=ChangePassword();`
	此 QuickScript 如果置于按钮上，或根据 “条件脚本” 或 “数据改本” QuickScript 来调用，则会打开一个对话框 （带可选键盘），提示用户输入当前口令和新口令，并验证新口令。

Logoff()

	从 InTouch 中注销用户。	
类别	安全性	
语法	`DiscreteTag = LogOff();`	
	参数	描述
	N/A	
附注	注销当前登录的用户，并将当前用户状态设置为默认值 “无” 操作员。	
示例	`DiscreteTag = LogOff();`	
另请参阅	**PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership(), AddPermission().**	

InTouch 安全系统标记

$OperatorName

类别	安全性
用法	如果采用基于操作系统或 ArchestrA 的身份验证方式，并且有人已经登录且尚未注销，则 $OperatorName 标记将包含操作员的全限定名称。否则，该标记将包含登录用户的名称 （与 $Operator 标记的内容相同）。
附注	N/A
数据类型	字符串
示例	`$Operator = "john";` `$OperatorName = "John Smith";`
另请参阅	**$Operator**

$OperatorDomain

类别	安全性
用法	如果选择了基于操作系统的安全性，并且操作员已成功登录，则 $OperatorDomain 标记将包含登录时指定的域名或机器名。如果用户在登录时选定了 Archestra 安全性，则 $OperatorDomain 将包含 "ArchestrA"。如果选定了 InTouch 安全性，则 $OperatorDomain 标记包含字符串 "InTouch"。如果选定 “无”，则它是一个空字符串 ""。
附注	N/A
数据类型	字符串
示例	`$Operator = "john";` `$OperatorDomain="CORPORATE_HQ";`
另请参阅	**$Operator**

$OperatorDomainEntered

类别	**安全性**
用法	只要 $PasswordEntered 标记发生变化，便会尝试从内部登录，而不显示任何图形用户界面。登录尝试将 $*Entered 标记用作输入用户名，将字符串值 $OperatorDomainEntered 用作域名（仅在当前模式是基于操作系统的安全性时使用）。如果安全模式并非基于操作系统，则会忽略此标记。
附注	N/A
数据类型	字符串

示例	`$OperatorEntered="john";`
	`$OperatorDomainEntered="Corporate_HQ";`
	`$PasswordEntered="password";`

另请参阅	**$Operator**

第 4 章

使用 InTouchView

按照其设计， InTouchView 是一套可视化用户界面产品，专门用在 ArchestrA Application Server 环境中。InTouchView 不使用单独的代码库，并且不要求在 InTouch 之外使用其它的安装程序。根据具体情况，应用程序开发人员会逐个指定应用程序将使用 InTouch 还是 InTouchView。开发人员使用 WindowMaker 中的设置，既可以将每个应用程序配置为功能齐备的 InTouch 应用程序，也可以将它们配置为 InTouchView 应用程序。

目录

关于 InTouchView

通过使用 InTouchView，您可以开发支持 ArchestrA 的应用程序，给操作员提供连接到 ArchestrA Application Server 系统时执行任务的功能。本节下面的部分介绍 InTouch 与 InTouchView 之间的差异。

InTouchView 与 InTouch 之间的差异

InTouchView 应用程序的主要不同在于，它们不具备连接到 ArchestrA Application Server Galaxy 之外的其它 I/O 数据源的功能。InTouchView 应用程序也不会生成报警，但它们支持显示与确认来自其它报警供应器的报警。InTouchView 应用程序使用 ArchestrA 安全模型。

启动 InTouchView

在您启动 WindowMaker 时， WindowMaker 会检查关于应用程序类型的文件信息。如果该信息表明该应用程序是 InTouchView 应用程序，则 WindowMaker 将相应限制应用程序的菜单与功能。

InTouchView 许可证

如果应用程序属 InTouchView 型，则 WindowViewer 要求 “许可证管理器”提供一个运行 InTouchView 应用程序的许可证。否则，它会要求提供一个运行全功能 InTouch 应用程序的许可证。

获取 InTouchView 许可证

- 通过 InTouch 或 InTouch TSE 启动 InTouch 应用程序。

在启动期间， InTouch 软件会评估正在启动的 “应用程序”。它会检查该应用程序是否为一个 InTouchView 应用程序，若确实如此，则它会要求 “许可证管理器”提供一个 InTouchView 许可证。

备注 如果该应用程序不是 InTouchView 应用程序，则该软件会要求 “许可证管理器”提供一个 InTouch 许可证。

如果为此应用程序成功读取了 InTouchView 许可证行，则 InTouch 会继续其启动过程。如果 InTouchView 许可证不可用，则 InTouch 软件要求提供一个 InTouch 许可证。如果成功读取 InTouch 行，则 InTouch 将继续其启动过程。否则， InTouch 会指出 InTouchView 许可证不可用，并提供一段简短的问题描述，然后提供退出、重试或按演示模式运行等选项。

创建新 InTouchView 应用程序

要运行 WindowMaker，您必须拥有一个 InTouch 许可证，不管您要创建的应用程序是 InTouchView 应用程序，还是全功能的 InTouch 应用程序。

要创建新的 InTouchView 应用程序

1. 指向**开始**、**程序** 及 **Wonderware**，并启动 InTouch。
2. 在 “InTouch 应用程序编辑器”中，指向**文件**，然后单击**新建**。您也可以单击工具栏上的**新建**图标。
3. 按照新建应用程序向导上的操作说明逐步执行。

4. 输入应用程序的名称及其描述，然后选中 InTouchView 应用程序复选框。



5. 单击**完成**。此时系统便会创建一个新的 InTouchView 应用程序，并将其安全类型设置为 ArchestrA。

运行 **InTouchView** 应用程序

您可以使用 InTouchView 或 InTouch 运行时许可证来运行 InTouchView 应用程序。

要在 InTouchView 中运行应用程序

- 在 WindowViewer 中打开 InTouchView 应用程序。

InTouchView 应用程序不会生成历史记录、报警，也不会连接到 Application Server Galaxy 之外的其它 I/O 客户端。它只会生成同 SYS 和 USER 相关的事件。

使用 **InTouchView** 菜单

您可以使用 InTouchView 菜单与功能来执行 InTouch 中可用的许多功能。不过， InTouchView 不支持有些功能。这些功能会出现在菜单中，但用灰白显示，以指出它们未被启用。下表专门介绍 InTouchView 中有哪些菜单与功能是不可用的。

InTouchView WindowMaker 菜单差异

下列**特别**菜单选项在 InTouchView 应用程序中不可用：

- 访问名
- 报警组
- 配置 .. 报警
- 配置 .. 历史记录
- 配置 .. 分布式名称管理器

下列选项在应用程序主体中不可用，并且在 InTouchView 应用程序中也不可用：

- 配置 .. 访问名
- 配置 .. 报警组
- 配置 .. 报警
- 配置 .. 历史记录
- 配置 .. 分布式名称管理器

下列**标记名字典**选项在 InTouchView 应用程序中不可用，并将以灰白显示：

- 报警
- 详细资料和报警
- 记录数据
- 记录事件
- 优先级 （用于记录事件）

转换 InTouchView 应用程序

通过使用 InTouch，您可以将应用程序从 InTouchView 转换到 InTouch 中，反之亦然。例如，如果您创建了一个 InTouchView 应用程序，但您需要该应用程序访问 Application Server 之外的其它数据源中的标记，则您可以将该应用程序转换为 InTouch 应用程序。如果您在 InTouch 中创建了一个应用程序，但您只想访问 Application Server，则您可以将该应用程序转换为 InTouchView 型。

将 InTouchView 应用程序转换为 InTouch 应用程序

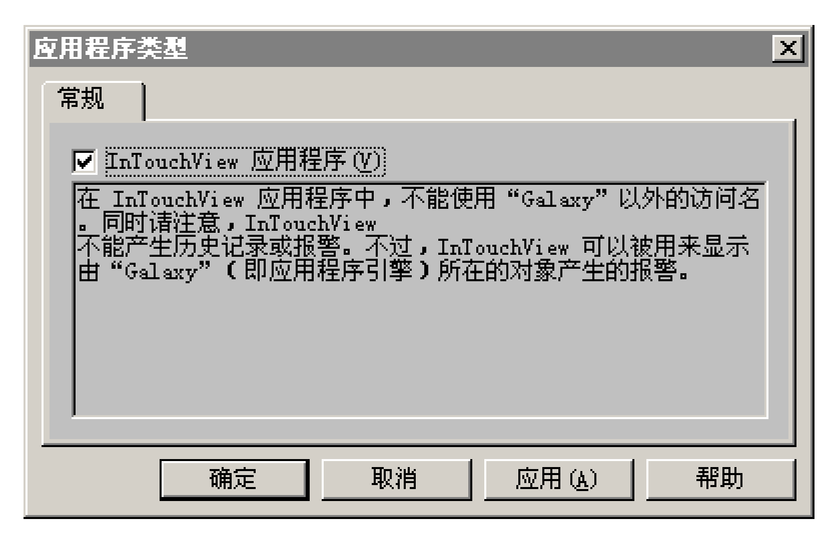
如果您需要应用程序访问 Application Server 之外的其它数据源中的标记，则可以将 InTouchView 应用程序转换为标准的 InTouch 应用程序。要运行从 InTouchView 转换为 InTouch 型的应用程序，要求使用完整的 InTouch 许可证。

备注 在 WindowViewer 仍处于活动状态时，您不能更改 “应用程序类型”。

要将 InTouchView 应用程序转换为 InTouch 应用程序：

1. 在 WindowMaker 中打开 InTouchView 应用程序。

2. 从**特别**菜单上，选择**应用程序类型**。此时会出现一个对话框，其中选定了 InTouchView 复选框。



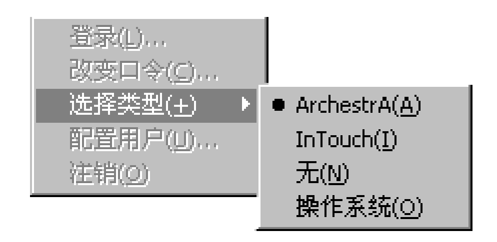
3. 取消选择该 InTouchView 复选框，然后单击**确定**。

将应用程序转换为 InTouch 型之后，您便可以使用曾在 InTouchView 应用程序中被禁用的那些功能。

备注 在将应用程序从 InTouchView 转换为 InTouch 型之后，您可能需要更改应用程序中的安全类型。

要更改安全类型：

在**特别**菜单上，指向**安全性**，然后指向**选择安全类型**。



4. 给应用程序选择安全类型。

   如需有关安全性的详细信息，请参阅使用 InTouch 安全性。

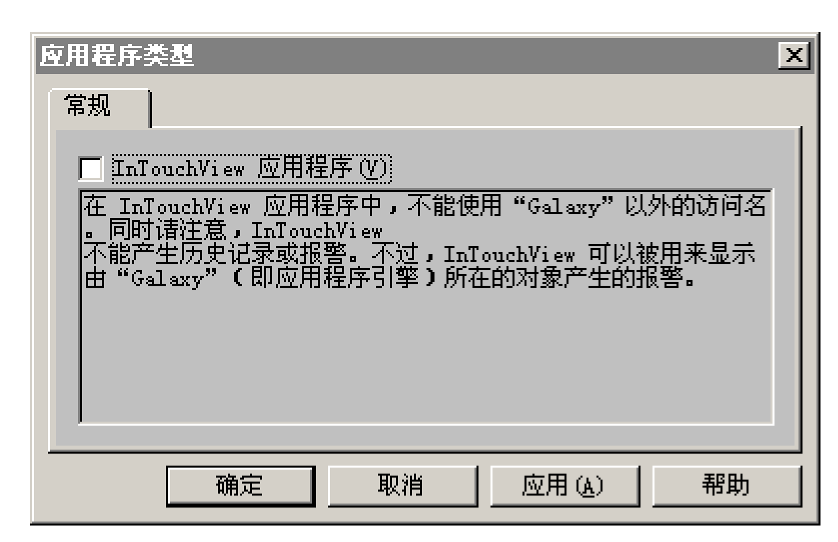
将 **InTouch** 应用程序转换为 **InTouchView** 应用程序

如果只想连接 Application Server，则您可以将 InTouch 应用程序转换为 InTouchView 型。

要将 InTouch 应用程序转换为 InTouchView 应用程序：

1. 在 WindowMaker 中打开 InTouch 应用程序。

   从**特别**菜单上，选择**应用程序类型**。此时会出现一个对话框，其中取消选定了 InTouchView 复选框



2. 选择该 InTouchView 复选框，然后单击**确定**。随后系统会检查应用程序是否使用了 Galaxy 之外的任何 “访问名”。如果确实如此，则会出现一个对话框，提醒您在切换到 InTouchView 之前需要删除所有非 Galaxy 的 “访问名”。系统还会提醒您 InTouchView 不支持历史记录或报警。

重要！ 在将应用程序更改为 InTouchView 之前，您必须删除所有 Galaxy 之外的 “访问名”。



3. 单击**确定**。

在将应用程序转换为 InTouchView 之后，有些菜单功能便会被禁用。

第 5 章

建立分布式应用程序

InTouch 设计为可以同时支持独立式和分布式应用程序。独立式应用程序指的是每个监视系统只使用一个"操作员界面"（Operator Interface，简称 OI），如锅炉部件控制。独立式应用程序一般较容易配置，对联网要求很低甚至不要求联网，并且只需简单的维护。相反，分布式应用程序则要复杂许多，通常会涉及好几层网络。通常，分布式应用程序有一个中央开发工作站、中央数据存储区、许多与中央工作站进行交互且相互之间也会发生作用的*客户端*工作站。

InTouch 提供了许多功能，可极大简化分布式应用程序的建立和维护。其中最强大的一项功能是"网络应用程序开发"（Network Application Development，简称 NAD）。 NAD 允许多个*客户端*工作站共同拥有一个应用程序的备份，而不限制这个应用程序的开发。在应用程序发生变化时， InTouch NAD 还会向这些*客户*站提供自动通知。

本章介绍如何使用 InTouch 的分布式功能、您可以采用的各种结构以及它们各自的优劣。

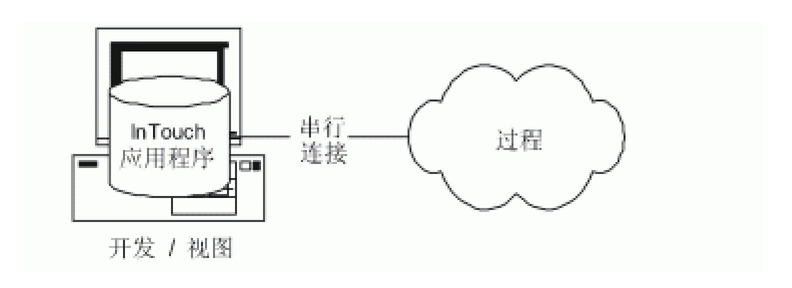
目录

网络结构

InTouch 是一套高度可配置的软件包，您可以根据应用程序的需要，采用各种方式灵活地进行设置。本节综述 InTouch 提供的各种结构及其各自的优劣。同时还将论及报警和历史等各种应用程序组件，对于这些系统，我们将在相应的章节中作更为深入的介绍。

独立式应用程序

独立式应用程序被定义为对每个监视进程只使用一个操作员界面的应用程序。这些应用程序通常由一个充当主操作员界面 （Operator Interface，简称 OI）的未联网个人计算机 （Personal Computer，简称 PC）组成。这台计算机直接连接 （例如使用串行电缆）工业流程。



在这种结构中，计算机上只安装单个 InTouch 应用程序。如果需要进行开发工作，则可以直接在这台计算机上开发应用程序。它还可以复制到另一台计算机上、进行修改，然后再复制回原来的计算机。独立式结构虽然不是一种网络结构，但为了完整起见，这里还是一并向您介绍。

优点

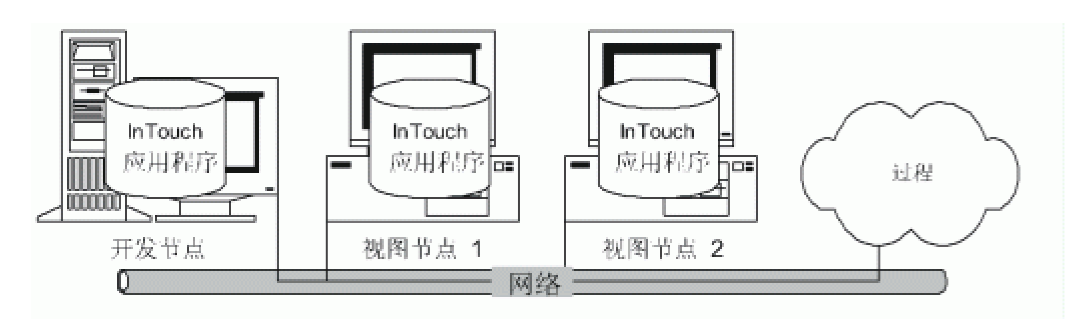
- 易于维护

缺点

- 局限于单个节点

基于客户端的结构

基于客户端的结构是第一种网络结构，它直接从独立式结构发展而来。它为运行 WindowViewer 和 NetDDE （View 节点）的每台计算机提供一个 InTouch 应用程序的独特备份。此应用程序可以安装到每台计算机的硬盘，或网络服务器上的一个特定位置。在下例中，应用程序将在开发节点上进行开发和测试，然后再复制到每个 View 节点。



每个 View 节点均具有相同的应用程序备份，因此它们对应用程序引用的任何数据源的访问也必须完全相同。这些数据源可以是 “I/O 服务器”、SQL 数据库、DOS 文件等。如果使用一个中央数据源（如网络共享式 “I/O 服务器”），则每个 View 节点需与共享服务器保持一个单独的对话，这就增大了网络负载。因此，如果预计网络繁忙，最好考虑在每个节点上使用单独的 “I/O 服务器”。

在应用程序维护方面，基于客户端的结构有利也有弊。由于每个节点均有各自的应用程序备份，开发节点对该应用程序的编辑功能没有作出限制。在该节点上，可以执行修改与测试，而不至影响正在运行的进程。此方法的缺点在于，将修改后的应用程序发布到各个 View 节点会比较费力。每个 View 节点必须先在本地关闭，然后将新的应用程序复制到其中，然后再重新启动各个 View。

优点

- 应用程序开发不受限制
- 每个节点均可自足，因此具有内在的冗余特性。
- 可使用的 View 节点数目不受限制。

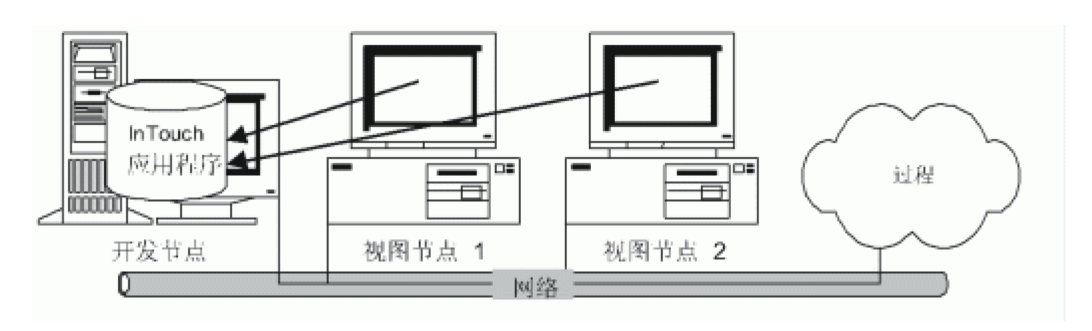
缺点

- 分布应用程序比较困难一些
- 所有节点对相同的数据源均须进行相同的访问

备注 此结构现已由 NAD 结构取代，在本章稍后部分将会介绍该种结构。在这里提到它的目的仅是为了让您对网络结构有一个完整的概念。

基于服务器的结构

基于服务器的结构允许多个 View 节点共享一个公用的 InTouch 应用程序。在下例中，两个 View 节点均从开发节点访问同一应用程序。每个 View 节点必须在网络软件中创建一个逻辑驱动器，并将它映射到开发节点的共享网络驱动器上。每个 View 节点还必须向 InTouch 程序注册共享的应用程序。



与基于客户端的应用程序一样，每个 View 节点对应用程序引用的任何数据源的访问必须完全相同。通过使用脚本组合来获取节点名，并根据该名称来更改每个数据源的位置，您可以使用多种方法自定义数据源的位置。

如需有关详细信息，请参阅 “InTouch 公共数据源配置”。

这种结构允许在应用程序发生变化及 WindowViewer 重新启动时更新 View 节点。

优点

- 只需维护单个应用程序
- 应用程序发生变化时， View 节点会自动更新

缺点

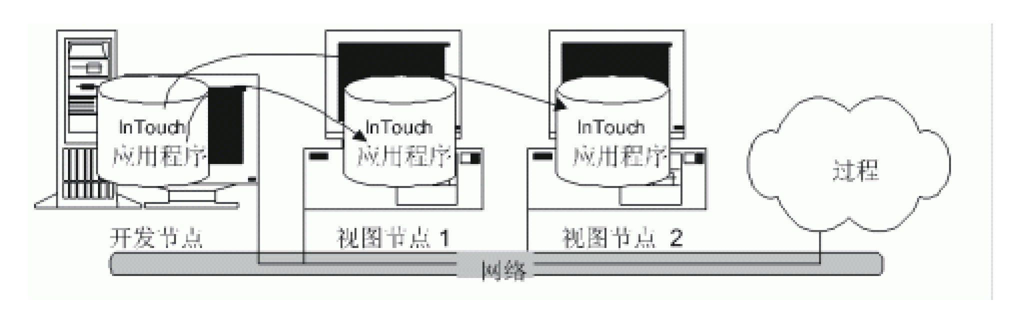
- 应用程序开发受到限制
- 如果 “开发”工作站崩溃，则没有冗余备份可供使用
- 所有节点必须使用相同的屏幕分辨率

备注 此结构现已由 NAD 结构取代，在本章稍后部分将会介绍该种结构。在这里提到它的目的仅是为了让您对网络结构有一个完整的概念。

网络应用程序开发 (NAD)

“网络应用程序开发”（简称为 NAD）是一个兼备基于客户端和基于服务器结构之优点的结构。NAD 不仅提供了就应用程序改变发出自动通知的功能，而且还提供了自动将更新的应用程序发布到各个 View 节点的功能。

在 NAD 结构中，应用程序的主备份在中央网络位置维护。与在基于服务器的结构中那样，每个 View 节点均会加载该网络应用程序，但它不是从服务器运行该应用程序，而是将应用程序复制到用户定义的位置并从那里运行。这就提供了基于客户端的结构所具有的冗余优点。在下例中，两个 View 节点都在开发节点上注册了主应用程序，但实际上是从其各自的硬盘运行它。



View 节点复制并运行主应用程序时，它会自动监视主备份中的变化。这些变化由主应用程序目录中的一个标帜指出。在编辑应用程序的同时，开发人员可以使用 WindowMaker **特别**菜单上的**通知客户**命令来手动设置此标帜。此标帜发生改变时，每个 View 节点将由一个用户自定义的动作指定该节点应做出的反应。这可以是忽略该标帜、自动关闭并重新启动 View 节点从而重新加载主应用程序。

备注 如果配置为将历史数据写入主应用程序节点的 “应用程序目录”，则所有 NAD 节点会尝试将它们的历史数据写入主应用程序。为避免出现此种情形，需要在每个 NAD 节点上将历史数据配置为写入本地目录，**而不是**主应用程序节点。

优点

- 只需维护单个应用程序
- 应用程序发生变化时会自动通知 View 节点
- 每个 View 节点都可以自定义对应用程序更新所作的反应
- 应用程序开发不受限制

缺点

- 如果将一个复杂的大型应用程序发布到许多节点，则在最初下载时，系统响应时间会明显拖长，更新会被优化
- 限制了在不同节点上运行不同应用程序的灵活性
- 对于较慢的网络或串行连接，应用程序传输可能会是问题

配置网络资源

InTouch 为分布式应用提供了许多可配置的选项。本节提供了一些示例，向您介绍如何设置和使用这些选项来发布 InTouch 应用程序。

配置文件的 UNC 路径

InTouch 支持给应用程序目录项、配置项目及分布式报警使用 “通用命名惯例”（Universal Naming Convention，简称 UNC）。UNC 允许直接访问基于网络的文件而不必创建映射驱动器。每个 UNC 地址由三个部分组成：**\\Node\Share\Path** 形式的 “节点”、“共享”和 “路径”。“节点”指的是包含文件共享的计算机节点名。“共享”指的是指定给该计算机上共享目录的逻辑名称。“路径”指的是跟共享有关的该文件的普通 DOS 路径。

备注 使用 Wonderware 的 SuiteLink 协议时， **NodeName** 长度限制为 15 个字符。

在通过 UNC 访问文件之前，您必须在需要访问的计算机上创建文件共享。您可以共享整个驱动器，或只共享它的一个目录；甚至您还可以定义该共享的安全设置。无论采用何种方法，您均须创建将在 UNC 地址中使用的共享名。如需有关创建文件共享的详细信息，请参阅 Windows 操作系统的手册。

在创建共享之后，您可以使用 UNC 地址在通常需要输入文件路径的地方引用该驱动器。例如， InTouch 程序允许给 InTouch 应用程序的位置输入 UNC 路径或标准 DOS 路径。

例如，假定您有一台网络名是 "EngineRm" 的计算机，并且您使用共享名 "Root" 共享了该计算机的根驱动器 "C:\"。要设置一个到 "C:\InTouch.32\Apps\Boiler" 应用程序的 UNC 路径，则必须使用下面的 UNC：

\\EngineRm\Root\InTouch.32\Apps\Boiler

如果 "Boiler" 目录自身已作为 "Boiler" 共享，则 UNC 可以简化为：

\\EnginerRm\Boiler

如果该共享是路径，则要求提供路径。

备注 如果需要写入一个由 UNC 地址引用的文件，则该共享必须是一个可读 / 写共享，即使它位于本地节点。如果您创建一个口令保护的共享，则除非您首先设置了一个网络驱动器映射，否则将不能使用 UNC 来访问该共享。您可以使用 Windows Explorer 从远程节点设置驱动器映射。

Wonderware SuiteLink 通讯协议

Wonderware 的通讯协议 SuiteLink 随 Wonderware FactorySuite 一起提供。Wonderware SuiteLink 使用基于 TCP/IP 的协议。 SuiteLink 专门设计用于满足象数据完整性、高吞吐量以及方便诊断这样的工业需要。 Microsoft Windows NT 和 Windows 2000 均支持这种协议标准。

SuiteLink 并非 DDE、 FastDDE 或 NetDDE 的替代品。客户端与服务器之间的每个连接均依赖于网络状况。 SuiteLink 是专为高速工业应用设计的，它具有下列特点：

“数值时间质量”（Value Time Quality，简称 VTQ）在所有传输给支持 VTQ 的客户端的数据值上贴上一个时间标签和质量指示器。

通过使用 Microsoft Windows NT 操作系统监视器，可以访问数据吞吐量、服务器加载、计算机资源消耗及网络传输等非常全面的诊断信息。此项功能对于规划和维护分布式工业网络而言至关重要。

不管应用程序是位于单个节点还是分布于多个节点，均可在应用程序之间可维持一致的高数据容量。

备注 网络传输协议是使用 Microsoft 标准 Winsock 接口的 TCP/IP。

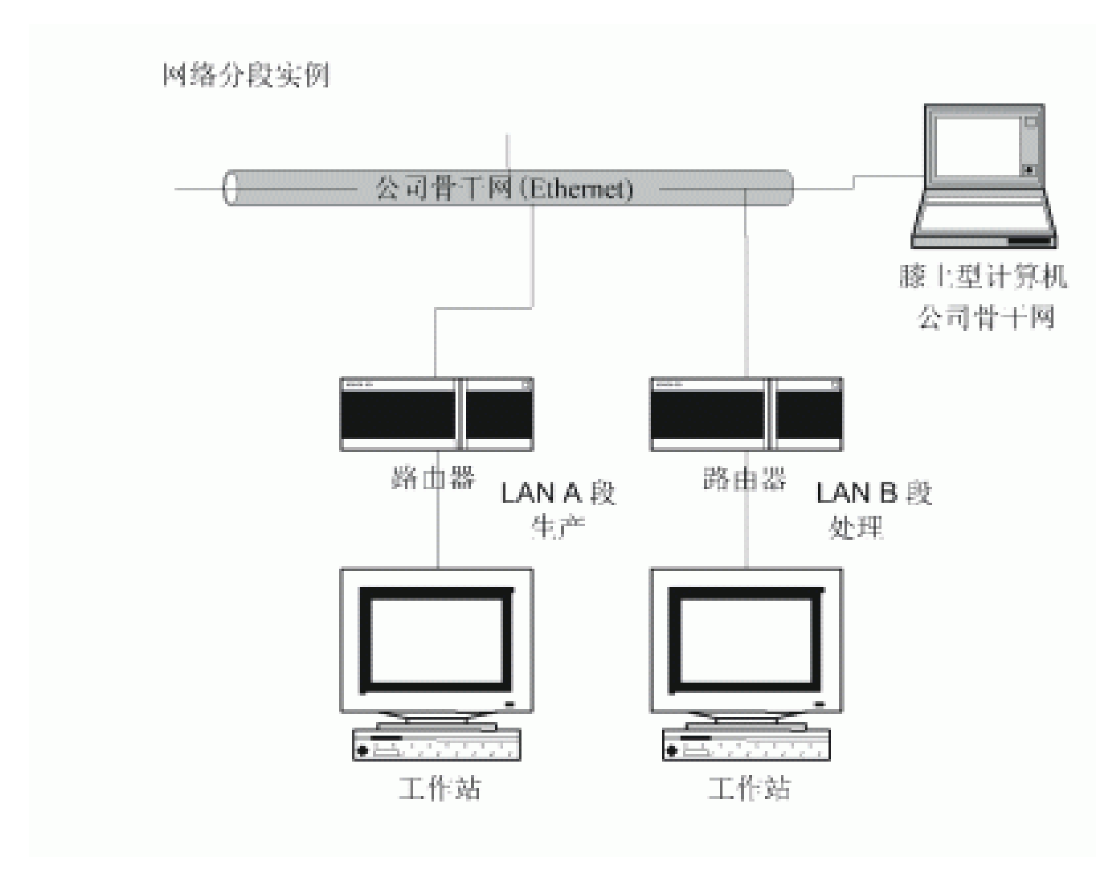
如需有关使用 SuiteLink 的详细信息，请参阅第 13 章 “I/O 通讯”。

排解网络疑难

在享受 InTouch 的分布式网络功能时，有一些关键的网络结构限制您必须予以考虑。在部署包含许多 InTouch 节点的大型网络时，首先要认识到并非所有的 “以太网”连接均是相同的。很多 MIS 部门会使用一种被称为 “路由器”的设备给网络分段。这些路由器扮演着交通警的角色，可以调节从一个 “以太网”网段到另一网段的传输流量。路由器具有“过滤”掉特定类型网络流量和地址的能力。

如果您使用的是 NetBEUI，则尝试连接另一建筑或不同城市的远程“I/O 服务器”会是一件让人失望的事情，您可能会发现它根本不起作用，即使网络已经连接妥当。如果发生此种情形，则通常是因为路由器被设置为过滤掉 NetBEUI 流量。一个解决办法是重新设定路由器，使之允许通过这种流量。不过，重新设定路由器有一个缺点，那就是现在 NetBEUI 广播会流向更多的网络用户，从而增加其它网段的流量负担。

备注 Wonderware 的 SuiteLink 协议不能与 NetBEUI 一起使用。SuiteLink 协议要求所引用的全部计算机均有一个名称，且其长度不得超过 15 个字符。



为避免出现此种情形，请换而使用 TCP/IP 协议，并将路由器设置为从一台路由器自动转接到下一台路由器。路由器必须配置为允许双向数据流动 （A 到 B 和 B 到 A），以最大限度降低从路由器进入其它网段的网络流量。通过使用 TCP/IP，您将可以使用很多种网络服务 （如 Internet、"帧中继"、 ISDN 等）进行远距离对话，从而提高网络性能。让您选用 TCP/IP 的另一个关键原因是， PLC 供应商们早已很快采用了 TCP/IP 这种协议。

Microsoft 将换而使用 TCP/IP 这样的工作变得非常容易。 Windows NT Server 附带针对 Windows 2000 或 Windows NT 4.x （或更高版本）的"网络客户端"软件，并提供内置的 TCP/IP 支持。 Windows NT Server 还具备管理动态分配 TCP/IP 地址和名称的功能。

Microsoft 的主要网络产品 Windows NT Server 也对联网的 InTouch 节点提出了一些要求。

例如，每当您试图通过另一个 "域"检索 InTouch 文件 （如历史记录文件）时，您必须提供口令并且必须在该域中有一个帐号才能建立连接。创建这些帐号以及频繁中断工作来输入口令使得检索操作几乎无法进行。此时，您可以使用两种办法来避免此问题：

1. 让所有的计算机加入相同的域；不过，域中的计算机太多也会引起管理方面的问题。
2. 在域之间建立 “信任关系”，允许一个域中的计算机共享另一个域的资源，而不必在每次进行连接时均须创建附加的帐号并提供口令。

明显可以看出，后一种设置更为可行，而且更容易设置和管理。此外，系统还提供其它高级的 “NT 域结构”模型，您需要根据类似的约束条件进行实施。

在这一类的情况下，建议您就整个网络实施的设计和拓扑结构问题咨询 Microsoft NT 系统专家。

InTouch 公共数据源配置

InTouch 允许您使用多种不同的结构来建立应用程序。无论采用何种结构，您均须慎重考虑应用程序将要访问的数据源及其访问方式。

每种结构都有一个共同的特性，在每个 View 节点上运行的应用程序如同是该节点所拥有的那样。初看起来这似乎不成问题，但是在您考虑 InTouch 应用程序可能包含的数据源引用时，它就会变得非常重要了。典型的数据源包括 “访问名”、 SQL 连接字符串或 “配方”文件。

这些数据源中的每种均是通过引用地址进行检索的，如 “配方”文件的 D:\PROCESS\RECIPE.CSV，或 SQL 连接的 DSN ＝ PROCDB。尽管对于在上面开发应用程序的计算机而言，这些地址具有意义，但对于没有 D: 盘或未注册 ODBC 数据源 PROCDB，且您要将该应用程序复制到其中并从中运行的 View 节点而言，它们则没有任何意义。

如果您打算将应用程序发布到多个节点，则需考虑数据源地址的影响。为此，您可以采用两种基本的方式：

1. 在每个 View 节点上创建数据源的相同备份，或
2. 对数据源只使用全局地址。

下面几节介绍这些选项，它们与两个主要的数据源有关：远程数据和文件访问。

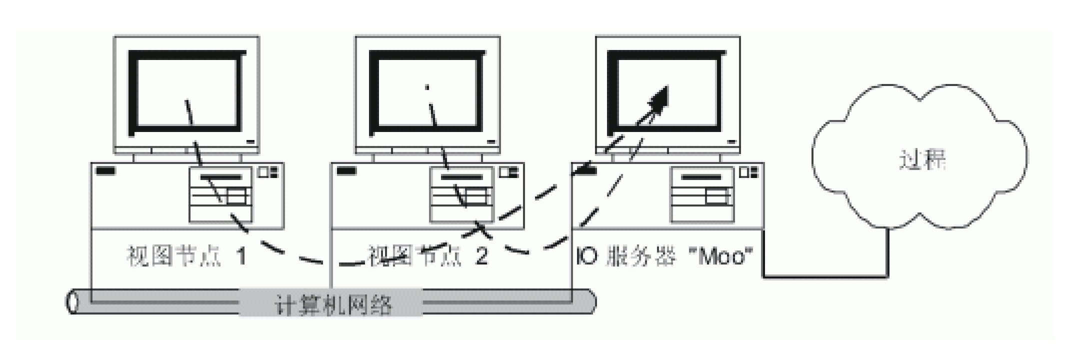
InTouch 访问名

InTouch 使用 “访问名”来引用实时 I/O 数据。每个 “访问名”均相当于一个 I/O 地址，可包含 “节点”、“应用程序”和 “主题”。在分布式应用程序中，可以将 I/O 引用设置为指向网络 “I/O 服务器”的全局地址，也可以将它设置为指向本地 “I/O 服务器”的本地地址。

如需有关 “访问名”的详细信息，请参阅第 13 章 “I/O 通讯”。 "

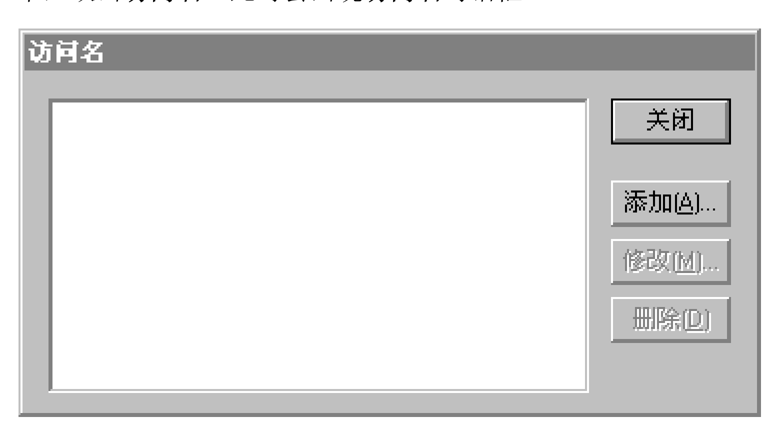
指向 I/O 数据源的全局地址

指向 I/O 数据的全局地址允许所有 View 节点共享一个基于公用网络的“I/O 服务器”。这可以节省使用多个“I/O 服务器”所需开销，但容错能力会下降，并可能导致整体性能降低。在下例中，两个 View 节点分别运行相同应用程序的一个备份，它们均引用相同的 I/O 数据源。由于每个应用程序均使用该数据源的全限定 I/O 地址，因此所有的引用均指向同一个“I/O 服务器”。



要设置此项配置

1. 在**特别**菜单上，单击**访问名**，或者在**配置**下的“应用程序浏览器”中，双击**访问名**。此时会出现**访问名**对话框。



2. 单击**添加**。此时会出现**添加访问名**对话框。



3. 在**访问名**框中，输入 **PLC1**。
4. 在**节点名**框中，输入 **Moo**。（不要在节点名前面加上 \\）。
5. 在**应用程序名**框中，输入 **Genius**。
6. 在**主题名**框中，输入 **PLC1**。

备注 您可以定义任意**访问名**，但**节点名**、**应用程序名**以及**主题名**必须引用安装了“I/O 服务器”的计算机。不过，为简便期间，建议您使用与“主题名”相同的名称。

7. 选择所使用的协议。
8. 单击**确定**。此时会再次出现**访问名**对话框，并在列表中显示新的“访问名”：

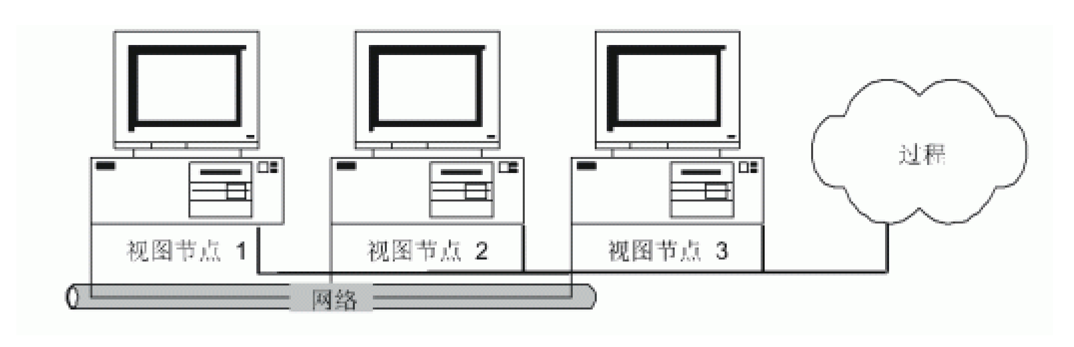


9. 单击**关闭**。

指向 I/O 数据源的本地地址

每个 View 节点均有自己的 “I/O 服务器”时，您可以使用指向 I/O 数据源的本地地址。这种结构能够提供容错操作，当网络中断时每个 View 节点均可独立运行。在下例中，两个 View 节点分别运行相同应用程序的一个备份，并引用其各自的 I/O 数据源。由于每个应用程序均使用该数据源的本地 I/O 地址，因此每个引用均指向本地 “I/O 服务器”。

不过，使用这种方法会显著增加过程网络的负载。这就是说，因为必须单独处理每个节点的请求，所以三个节点产生的流量将三倍于单个节点所产生的流量。



要设置此项配置

1. 在**特别**菜单上，单击**访问名**，或者在**配置**下的 “应用程序浏览器”中，双击**访问名**。此时会出现**访问名**对话框。



2. 单击**添加**。此时会出现**添加访问名**对话框。



3. 在**访问名**框中，输入 **PLC1**。
4. 将**节点名**框保留为空白。
5. 在**应用程序名**框中，输入 **Genius**。
6. 在**主题名**框中，输入 **PLC1**。

备注 您可以定义任意**访问名**，但**节点名**、**应用程序名**以及**主题名**必须引用安装了 “I/O 服务器”的计算机。不过，为简便期间，建议您使用与 “主题名”相同的名称。

7. 选择所使用的协议。
8. 单击**确定**。此时会再次出现**访问名**对话框，并在列表中显示新的 “访问名”：



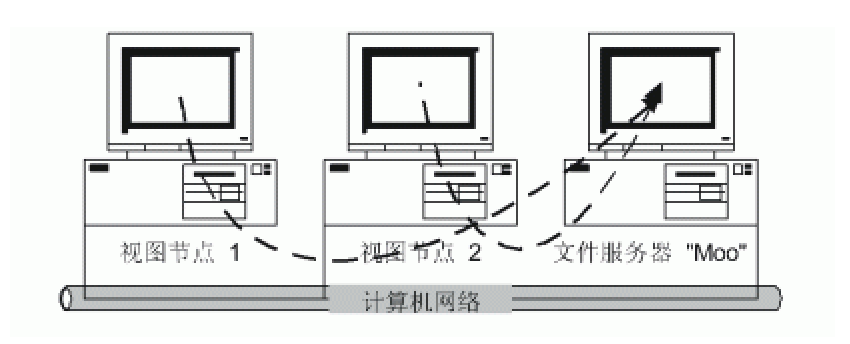
9. 单击**关闭**。

文件访问

InTouch 使用 DOS 文件、 FAT 或 NTFS 来读取和写入引用数据。某些程序 （如 Recipe Manager）会非常频繁地调用文件。在分布式应用程序中，文件引用可以设置为指向网络文件服务器的全局地址，也可以设置为指向本地文件的本地地址。

指向文件数据源的全局地址

指向文件数据的全局地址允许所有的 View 节点共享一个基于公用网络的文件集。这就提供了文件的单源维护，但它的容错性能要比本地备份差。在下例中，两个 View 节点分别运行相同应用程序的一个备份，它们可以引用相同的 “配方”文件。由于每个应用程序均使用映射成该文件全限定网络路径的盘符，因此所有的引用均指向相同的文件。



要设置此项配置

将网络驱动器映射到包含被引用文件的共享路径。在检索 “配方”文件的脚本中，输入下列语句：

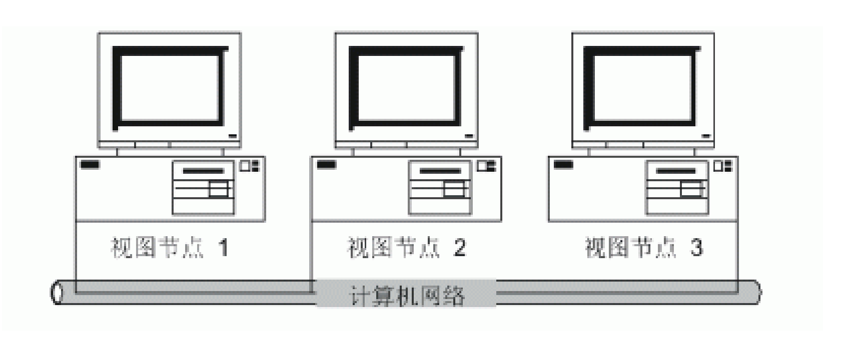
```
RecipeSelectRecipe("G:\Directory\Recipe.CSV", "review",
    "RecipeName");
```

其中 "G:\" 是指向 **Moo****Share** 的映射盘符。每个 View 节点必须单独予以配置，以映射这个相同的 "G:\" 驱动器。

如需有关详细信息，请参阅 “配置文件的 UNC 路径”。

指向文件数据源的本地地址

每个 View 节点自己包含一份该文件的备份时，可以使用指向文件数据的本地地址。由于网络中断时每个 View 节点均可独立运行，因此这种结构可提供容错操作，但它要求将文件的任何变化均复制到所有的 View 节点。在下例中，三个 View 节点分别运行相同应用程序的一个备份，并且它们都引用自己的 “配方” 文件备份。因为每个应用程序均使用该文件的本地地址，所以每个引用都指向本地文件。



要设置此项配置

使用本地地址 （例如 **C:\Directory**）直接引用文件。在检索 “配方” 文件的脚本中，输入下列语句：

```
RecipeSelectRecipe("C:\Directory\Recipe.CSV", "review",
    "RecipeName");
```

其中 "**C:**\" 是本地驱动器。

"Recipe.csv" 文件的一个备份必须存储在每部机器的本地目录 "**C:**\directory" 中。如果修改了该文件，则必须再次将它复制到每部机器。由于此种配置的维护工作相当困难，因此任何文件访问都只能是 “只读”，并且不允许修改本地文件。

InTouch 应用程序的 NAD 配置

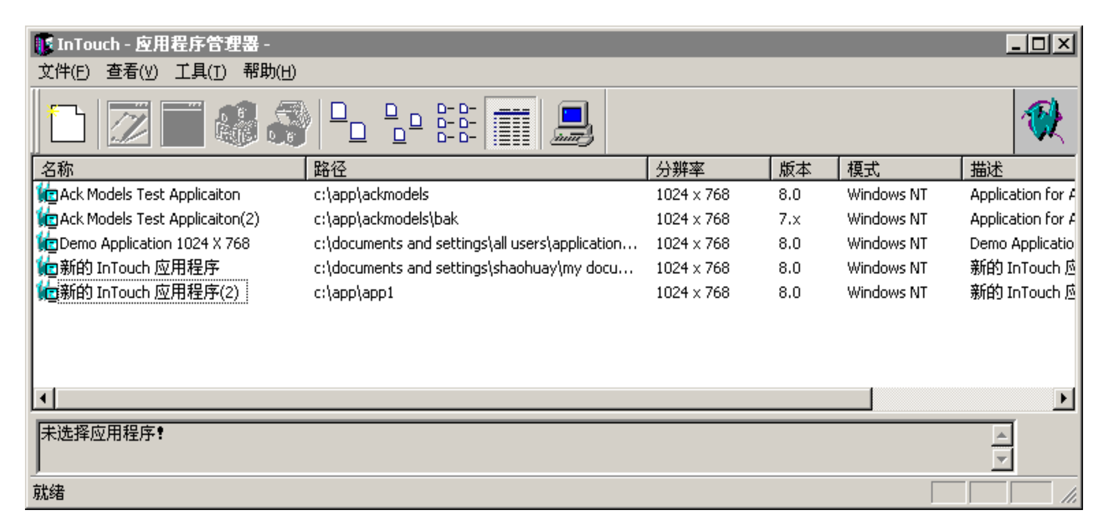
“网络应用程序开发”（简称为 NAD）是一个兼备基于客户端和基于服务器结构之优点的结构。NAD 不仅提供了就应用程序改变发出自动通知的功能，而且还提供了自动将更新的应用程序发布到各个 View 节点的功能。

备注 如果您将 WindowViewer 用作 NT 服务，则不能使用 NAD 功能。

如果配置为将历史数据写入主应用程序节点的 “应用程序目录”，则所有 NAD 节点均会尝试将它们的历史数据写入主应用程序。为避免出现此种情形，需要在每个 NAD 节点上将历史数据配置为写入本地目录，**而不是**主应用程序节点。

要配置应用程序的 **NAD**

1. 启动 InTouch 程序 (intouch.exe)。此时会出现 **InTouch 应用程序管理器**对话框。



2. 单击**节点属性**工具，或者在**工具**菜单上，单击**节点属性**。此时会出现**节点属性**对话框，并显示**应用程序开发**属性页。

提示 要快速访问**节点属性**对话框，请右击显示窗口的空白区，然后单击**节点属性**。



备注 **应用程序开发**属性页提供了多个选项，供您指定 NAD 的工作方式。这些设置是在每个 View 节点上配置的，而不是开发节点上配置的。这样每个 View 节点均可以使用独特的配置。

此外，在您将 WindowViewer 当作 NT 服务运行时，它允许通过操作系统登录和注销 （例如操作员换班）时来连续操作 WindowViewer。通过选择此选项，您还可以让 InTouch 在电源故障或关机并重新开机后自动启动。如此一来，就可以实现 WindowViewer 无人值守式自动启动，同时又不会影响到 NT 操作系统的安全性。不过，如果您将 WindowViewer 当作 NT 服务运行，则不能使用各项 NAD 功能。

如需有关作为 Windows NT 服务运行的详细信息，请参阅附录 A “InTouch Windows NT 服务综述”。

3. 选择**允许网络应用程序开发**以启用 NAD。此时将激活五个**改变模式**选项。每个节点只能选择一种改变模式。

备注 初次复制主应用程序可能要比后续的更新花费更长时间。

4. 在**本地工作目录**框中，输入要 WindowViewer 将主应用程序复制到其中的目录。

   如果这是开发节点，则可输入本地目录路径，如 **\c:\InTouch\NAD**。您也可以输入网络远程 UNC 路径，如 **\\node\share\path**。这对基于文件服务器的网络来说是比较方便的，在这样的网络中，大多数文件均存储在一个中央位置。如果这是客户端节点 （仅限运行时），则可以使用本地目录路径。如果未指定目录，则 WindowViewer 会在从中启动 WindowViewer 的目录中自动创建一个子目录 "NAD"。

   建议您尽量使用本地目录，以避免因网络延迟和故障而影响 WindowViewer 的操作。

   注意！ 不要使用 “根”目录或指向根目录的 UNC 路径名。在复制主应用程序目录之前， View 节点将删除指定的目标应用程序目录中的所有文件及子目录。因此，<u>*切勿*</u>使用主应用程序目录的路径或指向主应用程序目录的 UNC 路径。

   此目录应视作临时目录，并且除了由 NAD 自身复制的文件之外，不得将其它文件保存到其中。

   如需有关 UNC 路径的详细信息，请参阅 “配置文件的 UNC 路径”。

5. 设置**轮询周期 （秒）：**时间。此值设置 View 节点轮询开发节点以检查是否发生变化的时间间隔 （以秒计）。

备注 指定此项设置时，请务必小心。如果将此值设置得过小，WindowViewer 需浪费大量时间来检查主应用程序变化。这会干扰 WindowViewer 为运行中的应用程序提供服务。

6. 在**改变模式**组中，为在主应用程序发生变化时 WindowViewer 要采取的动作选择选项。

忽略变化	让运行时 (WindowViewer) 节点忽略开发节点上的任何变化。
重新启动 WindowViewer	运行时节点复制更新的主应用程序（如果已配置为这样），然后在运行时节点上重新启动 WindowViewer。
提示用户去重新启动 WindowViewer	显示一个交互式消息框，通知操作员应用程序已发生改变，并询问是否要重新启动 WindowViewer。
装载变化到 WindowViewer	将开发节点上发生的变化动态加载到 WindowViewer 中。对于较大规模的更新，这样可能会影响性能。
提示用户将变化加载到 WindowViewer 中	显示一个交互式消息框，通知操作员应用程序已发生变化，并询问是否要将变化动态加载到 WindowViewer 中。

如需有关详细信息，请参阅 “自定义 NAD 更新功能”。

7. 单击**确定**。

自定义 NAD 更新功能

除上述五个更新选项外， NAD 还提供了下面这些工具，供您自定义应用程序的更新行为：

工具	描述
$ApplicationChanged	在主应用程序发生改变时提供提示。此标记名可用于触发消息，以告诉操作员主应用程序已发生改变。您也可以在数据改变脚本中使用 **$ApplicationChanged** 系统标记名，以建立节点更新通知脚本。此脚本可以包括启动自定义对话框或关闭特定的进程。随后，您可以使用 **ReloadWindowViewer()** 来初始化更新过程。
ReloadWindowViewer()	执行此 QuickScript 函数时，它会使用更新的主应用程序去动态更新 WindowViewer，同时又不中断服务。例如，您可以在某个按钮上使用此函数，让操作员选择在适当的时间更新应用程序。此函数也可用在 QuickScript 中，以便在指定的时间或在换班时自动更新。
RestartWindowViewer()	执行此 QuickScript 函数时，它会自动关闭 WindowViewer，复制更新的主应用程序（如果已设置为如此），然后重新启动 WindowViewer。 **备注**此函数已由上述 **ReloadWindowViwer()** 函数取代。

备注 要使用这些函数，必须将**节点属性**对话框中的**改变模式**选项设置为**忽略变化**。设置此选项可防止系统受自定义 NAD 功能的干扰。

如需有关这些功能的附加信息，请参阅联机 *InTouch 参考指南*。

手动通知客户应用程序改变

在应用程序开发期间，您可以执行 WindowMaker **特别**菜单上的**通知客户**命令，以自动更新 InTouch 客户端应用程序。

执行此项命令时，系统会设置一个标帜，向所有的远程 View 节点通知主应用程序已经发生改变。于是这些客户端可以根据为每个节点定义的参数而自动启动更新过程。

提示 您也可以使用 InTouch QuickScript 来通知客户端。

如需有关详细信息，请参阅第 12 章 “实时趋势和历史趋势”。

应用程序复制过程

WindowViewer 节点复制应用程序时，它会尽可能地在复制过程中保留主应用程序的属性 （只读、系统、隐藏等）。 WindowViewer 还会复制主应用程序的所有文件及子目录。复制过程不会复制以下文件：***.WVW**、***.DAT**、***.LGH**、***.IDX**、***.LOG**、***.LOK**、***.FSM**、***.STG**、***.DBK**、***.CBK**、***.HBK**、***.KBK**、***.LBK**、***.NBK**、***.OBK**、***.TBK**、***.WBK**、 ***.XBK**、 ***.$$$**、 **RETENTIV.X**、 **RETENTIV.D**、**RETENTIV.A**、 **RETENTIV.S**、 **RETENTIV.H**、 **RETENTIV.T**、**WM.INI**、 **DB.INI**、 **LINKDEFS.INI**、 **TBOX.INI**、 **GROUP.DEF** 以及 **ITOCX.CFG**。

备注 WindowViewer 会循环删除目标应用程序目录中的所有文件和子目录。此目录应制作临时目录 （不能存放文件）。

应用程序编辑锁定

InTouch 应用程序一次只能由一个开发人员进行编辑。为防止多个开发人员同时编辑某个应用程序，在编辑过程中 WindowMaker 会锁定该应用程序。如果您试图将应用程序加载到已创建了锁定的 WindowMaker，则会收到一则消息，告诉您该应用程序正在由另一台计算机编辑，因此您不能编辑它。消息中同时还会指明当前编辑该应用程序的节点的名称。

备注 如果 WindowMaker 在加载应用程序时异常关闭，则不会自动删除 appedit.lok 文件。不过，您可以通过从应用程序目录中删除 appedit.lok 文件来手动删除锁定。

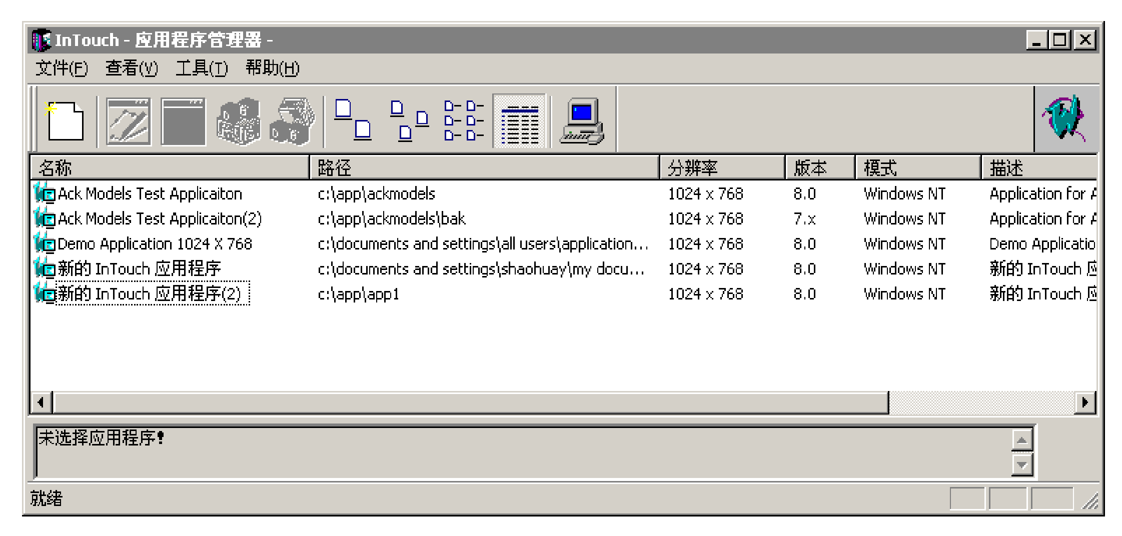
动态分辨率转换 (DRC)

“动态分辨率转换”（Dynamic Resolution Conversion，简称 DRC）与其它分布式功能一起，可使应用程序免受屏幕分辨率的限制。在 NAD 结构中，可以在开发节点上创建和维护 InTouch 应用程序，然后将它们复制到多个 View 节点。 DRC 允许在所有这些节点上查看应用程序，即使它们运行在不同的屏幕分辨率下。

DRC 可供每个 View 节点根据一系列自定义选项来缩放应用程序，包括自定义分辨率。此种比例调整可以在 WindowViewer 编译应用程序时进行，并且不需要 WindowMaker。每个 View 节点均可使用不同的 DRC 设置，因此每个 View 节点均须配置自己的设置。

要配置应用程序的 DRC

1. 启动 InTouch 程序 (intouch.exe)。此时会出现 **InTouch 应用程序管理器**对话框。



2. 单击**节点属性**工具，或者在**工具**菜单上，单击**节点属性**。此时会出现**节点属性**对话框，并显示**应用程序开发**属性页。

   提示 要快速访问**节点属性**对话框，请右击显示窗口的空白区，然后单击**节点属性**。

   备注 在 “应用程序管理器” 窗口中选定应用程序，然后从**文件**菜单上选择**属性**命令，以便显示所选应用程序的**属性**对话框。

3. 单击**分辨率**标签。



4. 如果希望 WindowViewer 根据所选的分辨率选项在本机上缩放主应用程序，请选择**允许 WindowViewer 动态改变分辨率**。（下面将介绍三个分辨率选项）。

备注 如果未选择此选项，则仅当节点屏幕分辨率与应用程序开发节点的屏幕分辨率完全相同时， WindowViewer 才会运行应用程序。如果分辨率不同， WindowViewer 会提示操作员运行 WindowMaker，以便将应用程序的分辨率转换为节点的分辨率。如果已设置指向主应用程序目录的 UNC 路径，则执行此项操作时请务必小心，因为这只会修改原始应用程序。

---

5. 如果希望 WindowViewer 以开发时的分辨率运行应用程序而忽略节点分辨率，请选择**使用应用程序分辨率**。例如，如果应用程序在 640x480 的分辨率下开发，而节点的分辨率是 1024x768，则 WindowViewer 将不会动态缩放应用程序。相反，该应用程序将在 640x480 的分辨率下显示。

6. 如果希望 WindowViewer 以节点分辨率运行应用程序而忽略开发应用程序时的分辨率，请选择**转换为屏幕视频分辨率**。例如，如果节点以 640x480 的分辨率运行，而应用程序是在 1280x1024 的分辨率下开发的，则 WindowViewer 会动态调整应用程序分辨率 （降低），使之适应节点的 640x480 显示器。（这会是最常用的设置）。

7. 如果希望 WindowViewer 以**象素宽度 (X)** 和**象素高度 (Y)** （必须是整型值）框中指定的分辨率运行应用程序，请选择**自定义分辨率**。应用程序的分辨率和节点的分辨率均会被忽略。例如，如果**像素宽度 (X)** 和**像素高度 (Y)** 分别被设置为 512 和 384，则应用程序将进行动态缩放，以适应节点显示器上的 512x384 象素区。

8. 单击**确定**。

使用多监视器系统

如今的市场上供应好几种高级图形适配卡，可供您同时将多个 VGA 监视器与系统相连。这些监视器互相配合，可以形成一个非常大的虚拟屏幕。例如，将四个 17 英寸的监视器堆叠成一个立体系统：两个在下，两个在上。由于每个屏幕的分辨率均为 800x600，因此所形成的虚拟屏幕分辨率高达 1600x1200 象素。

“动态分辨率转换”（Dynamic Resolution Conversion，简称 DRC）可以轻松支持这些多监视器系统。只需选择 DRC 分辨率转换选项，就可以充分利用整个虚拟显示器或仅使用其中的一部分。

如果应用程序调整为在偶数个监视器上运行，则显示的某些对话框覆盖多个监视器时会出现问题。其中的对话框之一**数字小键盘**就可能会因无法访问某些键而产生问题。为解决此问题， InTouch 提供了多个多监视器配置选项。

要在节点上配置多监视器设置

1. 使用合适的文本编辑器 （例如 Windows “记事本”）打开 Windows 目录中的 **WIN.INI** 文件。

2. 找到 **[InTouch]** 部分并添加下列参数：

3. [InTouch]

参数	描述
MultiScreen=1	打开多屏幕模式
MultiScreenWidth=640	单个屏幕的象素宽度
MultiScreenHeight=480	单个屏幕的象素高度

例如，如果您计算机的分辨率是 2560x1024，分开显示在两个并排的屏幕上，则可输入下列内容：

[InTouch]

MultiScreen=1

MultiScreenWidth=1280

MultiScreenHeight=1024

备注 上述各项会影响数字小键盘和 QWERTY 键盘。其它 InTouch 对话框和选项框不会受到影响。

分布式应用程序和时区

InTouch 提供的服务可以在跨多个时区的环境下方便地使用应用程序。这些服务由报警和历史系统使用，以允许按发生报警和事件的当地时间来查看数值。例如，如果加州的一名工程师正在观察上午 10 点发生在堪萨斯一个制造厂报警，工程师将看到报警发生时的加州当地时间，即上午 8 点。如果工程师在观察该工厂的历史数据，则情形也是如此。

这些服务的关键是将 “通用协调时间”（Universal Coordinated Time，简称 UCT ；也就是 “格林尼治标准时间，或简称 GMT）用作时间参考。每台计算机都配置有当地时间及其所在时区的 UCT 时间差。在上例中，加州计算机的时区设置为 GMT 8 点，而堪萨斯计算机的时区则设置为 GMT 6 点。

InTouch 使用这些 GMT 时差作为检索所有报警和历史数据的基础。在上例中，加州的 InTouch 应用程序从堪萨斯的应用程序收到报警时，它会同时查看两台计算机的 GMT 时差，以确定报警发生时的加州当地时间。因此，在 UCT 6 点时区上午 10:00 发生的报警等同于在 GMT 8 点时区上午 8:00 发生的报警。要使用此项功能，每台计算机均须配置有 GMT 时差。

如需有关详细信息，请参阅第 8 章 “历史趋势和夏令时”。

分布式报警

InTouch 支持 “分布式报警系统”。该系统可以显示由本地 InTouch 应用程序及其它网络 InTouch 应用程序的报警系统所产生的报警和事件。这些报警可以在本地 InTouch 节点或从网络上的远程节点进行确认。

如需有关设置和配置分布式警报系统的详细信息，请参阅第 9 章 “报警 / 事件”。

分布式历史

InTouch 提供的分布式历史记录系统可以从任何 InTouch 5.6 （或更高版本）应用程序检索历史数据，甚至是跨越网络的历史数据。此系统通过同时从多个历史数据库远程检索数据，扩展了标准的 InTouch 历史记录系统的功能。这些数据库被称为历史供应器。您最多可同时显示八个历史供应器，且每个供应器对应历史趋势图中的一支笔。

备注 历史供应器可以配置为本地 InTouch 历史或 IndustrialSQL(InSQL) 历史供应器。

如需有关设置和配置分布式历史系统的详细信息，请参阅第 12 章 “实时趋势和历史趋势”。

第 6 章

标记名字典

“标记名字典”（运行时数据库）是 InTouch 的核心。在运行时，数据库包含数据库中所有项目的当前值。为了创建运行时数据库， InTouch 需要了解所有要创建的变量的有关信息。您必须给每个变量指定标记名和类型。对某些类型的变量， InTouch 还需要一些额外的信息。例如，对于 I/O 型标记名， InTouch 需要更多的信息，以便能获取数据并转换它供内部使用。“标记名字典”就是用来输入这些如需有关详细信息的机制。

本章还将介绍 DBDump 和 DBLaod 这两个数据库实用程序。DBDump 可以将 InTouch 应用程序 “标记名字典”导出为一个文本文件，供其它软件包 （如 Microsoft Excel）访问，以进行修改、存储等。 DBLoad 可以将在其它软件包 （如 Excel）中创建的标记名数据库或将另一个 InTouch 应用程序中的 DBDump 文件加载到现有的 InTouch 应用程序中。

目录

- 显示标记名使用计数
- 替换标记名
- 转换占位符标记名
- 定标 I/O 标记名
- 内部系统 $Tagnames
- 标记名点域
- 标记名字典实用程序

标记名字典特殊功能

“标记名字典”提供了下列各项特殊功能：

功能	描述
标记浏览器	“标记浏览器”用于从 FactorySuite 应用程序或任何其它支持 “标记名字典”接口的标记源选择标记名和标记名**点域**、远程标记名引用，以及 SuperTag 成员标记名。
标记名交叉引用	“标记名交叉引用”可供您将标记名交叉引用至应用程序中用到它的一些位置，包括窗口、脚本、 SQL 配置、SPC 触发器等。您可以打印交叉引用信息或将它储存到文件。
SuperTag	InTouch 支持 SuperTag 结构，此结构可供定义复合型标记名。您可以定义包含多达 64 个成员标记名和 2 层嵌套的 SuperTag 模板。成员标记名的行为和普通标记名完全相同，它们支持趋势、报警和所有的标记名**点域**。
引用	远程标记名引用可供 InTouch 访问 I/O 服务器上的数据，而不必在本地 “标记名字典”中创建标记名。远程引用可供导入和导出窗口或 QuickScript 而不必执行从占位符到标记名的转换工作。
扩展的标记名支持	InTouch 在它的 “标记名字典”中能够支持多达 61405 个标记名。（系统可支持的标记名数目由软件许可证确定）。

标记名类型

在 InTouch 数据库中定义标记名时，您必须根据其用法为每个标记名指定一种类型。例如，如果标记名要读或写另一个 Windows 应用程序（如 “I/O 服务器”）中的值，则它必须是一个 I/O 型标记名。下面介绍每种 InTouch 标记名类型及其用法。

内存型标记名

内存型标记名存在于 InTouch 应用程序内部。您可以使用它们来创建系统常数和仿真。您还可以使用它们来创建供其它 Windows 程序访问的计算变量。例如，您可以使用初值 3.1416 来定义一个内存型标记名，也可以将配方储存至内存标记名组。在仿真中，您可以使用内存型标记名控制后台 QuickScript 脚本的动作。例如，您可以定义一个内存型标记名 "COUNT"，它会在某个动作 QuickScript 中改变，从而给某个过程的当前 “步骤”生成各种动画效果。“内存”型标记名共有四种：

内存离散型

内部离散型标记名，值为 0 (False, Off) 或 1 (True, On)。

内存整型

介于 -2,147,483,648 至 2,147,483,647 之间的 32 位带符号整数值。

内存实型

浮点（小数）内存型标记名。浮点数可介于 -3.4e^{38} 和 3.4e^{38} 之间。所有的浮点计算均按 64 位精度进行运算，但是会用 32 位来保存结果。

内存消息型

最长可达 131 个字符的文本字符串标记名。

I/O 型标记名

所有从另一个 Windows 程序读取值或将值写入另一个 Windows 程序的标记名均为 I/O 型标记名。这包括所有来自可编程控制器、过程计算机的输入和输出，以及来自网络节点的数据。 I/O 标记名可以通过 Microsoft “动态数据交换”（Dynamic Data Exchange，简称 DDE）或 Wonderware SuiteLink 通讯协议进行访问。

可读 / 写 I/O 型标记名的值发生改变时，它会被立即写入远程应用程序。在远程应用程序中，每当链接到标记名的项目发生改变时，该标记名也会从远程应用程序上更新。缺省条件下，所有的 I/O 标记名均被设置为可 “读 / 写”。不过，通过在**标记名字典**对话框中选择 “只读” 选项，您可以将它们限制为只读。 I/O 型标记名共有四种：

I/O 离散型

离散型输入 / 输出标记名，值为 0 (False, Off) 或 1 (True, On)。

I/O 整型

介于 -2,147,483,648 至 2,147,483,647 之间的 32 位带符号整数值。

I/O 实型

浮点 （小数）型标记名。浮点值可介于 $\pm 3.4e^{38}$ 之间。所有的浮点计算都按 64 位精度执行运算，但是会用 32 位来保存结果。

I/O 消息型

最长可达 131 个字符的文本字符串输入 / 输出标记名。

如需有关使用 I/O 标记名的详细信息，请参阅第 13 章 “I/O 通讯”。

其它类型的标记名

有多种特殊的标记名类型可供您指定给标记名，以执行复杂的功能，如创建动态报警显示、历史趋势、监测或控制每个历史趋势笔绘制的标记名。您还可以使用一些间接型标记名，将一个标记名赋给多个源标记名。这些特殊类型的标记名现介绍如下。

组变量

组变量型适用的标记名指定有 “报警组”，用以生成动态报警显示、磁盘记录和打印记录。您可以使用**组变量**型标记名来创建报警窗口或报警记录，以显示与特定的组变量关联的所有报警。您会可以通过将不同的 “报警组” 赋给**组变量**标记名，来控制所显示或记录的报警。

组变量型标记名也可以用来创建按钮，通过单击这些按钮，操作员可以在相同的报警窗口中有选择地显示工厂中不同区域的报警。与 “报警组” 关联的所有**点域**均可应用于**组变量**型标记名。

备注 “组变量”型标记名在 InTouch 7.11 中仍受支持，但它已经没有什么作用。因为不再使用“标准报警系统”，所以它不再用于动态改变“报警组”。

如需有关“报警”的详细信息，请参阅第 9 章“报警 / 事件”。

历史趋势

创建历史趋势时， InTouch 需要用到**历史趋势**型标记名。与历史趋势关联的所有**点域**均可应用于**历史趋势**标记名。

标记 ID

这是与历史趋势对象一起使用的特殊标记名类型。您可以使用**标记 ID** 型标记名来检索在历史趋势上绘制的标记名的有关信息。在大多数情况下，您会使用**标记 ID** 型标记名来显示赋给指定笔的标记名的名称，或用以改变赋给该笔的标记名。

您可以在 QuickScript 中使用一条语句，将新标记名赋给任意历史趋势中的任意一支笔。例如，您可以在 QuickScript 中使用下列语句：

```
MyHistTrendTag.Pen1=MyLoggedTag.TagID;
```

执行此 QuickScript 时，历史趋势中与**历史趋势**标记名 "MyHistTrendTag" 关联的笔 Pen1 将开始为 "MyLoggedTag" 绘制历史记录数据趋势图。

如需有关使用**标记 ID** 型标记名的详细信息，请参阅 *InTouch 参考指南*。

间接离散、间接模拟、间接消息

间接型标记名可供创建一个窗口，并将该窗口中的标记名重新指定给多个源标记名。例如，假设有十五个完全相同的泵，您想要让操作员监视它们的报警条件。此时，您不必创建十五个不同的窗口（每个泵一个），而只需在窗口中使用间接型标记名，然后用 QuickScript 将间接型标记名指向已进入报警状态的那个泵所对应的源标记名，此时即可调用与各个泵关联的源标记名。您也可以使用“触动按钮” QuickScript，让操作员手动选择要显示的泵。此种方法可减少开发时间和应用程序大小。

备注 InTouch 还支持间接 SuperTag。所有间接 SuperTag 显示在“标记名字典”的**标记名类型**对话框中。间接 SuperTag 也可用在 InTouch QuickScript 中。

如需有关详细信息，请参阅“创建间接 SuperTag”。

将某个间接型标记名设置为等于另一个源标记名时，会让这个间接型标记名与源标记名在所有方面均完全相同，其中包括点域、脚本等。如果源标记名的值发生改变，则间接型标记名会反映这种变化。如果间接型标记名的值发生改变，则源标记也会相应改变。您可以在数据库中将间接标记值定义为可保留，并在启动时将它们重置为上一次标记名赋值时的状态。

间接标记名通过使用 **.Name** 点域进行赋值。例如，如果创建了一个间接模拟型标记名 "Setpoint"，并在 QuickScript 中使用下面的表达式，则 "Setpoint1" 将变成值 "Setpoint" 的源标记名：

```
Setpoint.Name = "Setpoint1";  或 Setpoint.Name =
    Setpoint1.Name;
```

您也可以串联标记名以便在间接标记名中使用。例如，如果创建了一个 “数据改变” QuickScript，让它在每次标记名 "Number" 的值发生变化时执行，则间接标记名 "Setpoint" 也会相应地发生改变：

```
Number=1;

Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

此 QuickScript 执行时，模拟型标记名 "Number" 的值会转换为文本，并串联到模拟型标记名 "Setpoint"，从而使 "Setpoint.Name" 等于 "Setpoint1"。间接模拟型标记名可用于整型 （整数）和实型 （浮点数）这两种标记名。

调用源标记名并使用英文引号括起源标记名时：

```
Indirect.name= "mytag"
```

或者，源标记名是通过串联文本字符串与变量来定义的：

```
Indirect.Name = "mytag" + Text(Number, "#");
```

在这种情况下，源标记名对于间接型标记名而言处于不活动状态。在第二次执行 QuickScript 之前，间接型标记名将不能引用源标记名的特性。我们建议您去确认是否所有间接型标记名引用的均是活动标记名。

只有在下列条件下，源标记名才会被激活：

- 显示在一个打开的应用程序窗口中
- 用在 “窗口”或 “动作” QuickScript 中，并且与该脚本关联的应用程序窗口已打开
- 当前正由 “实时趋势”使用
- 用于报警目的
- 作为事件来记录并且已启用 “事件记录”功能
- 记录到 “历史记录”中，并且已启用 “历史记录”功能
- 用在 “键”、“条件”、“数据改变”或 “应用程序” QuickScript 中
- 作为 InTouch SPCPro 中的 “自动采集”标记名
- 当前正由客户端应用程序 （如 Microsoft Excel）使用 DDE 进行访问

- 由本地客户端使用 “点存取” (PTACC.DLL) 进行引用
- 赋给 （或由之引用）“间接标记”（因为脚本必须请求 I/O 点的正确值，所以会出现瞬间缓迟）

或

- “InTouch 访问名”被配置为供 DDE 协议**提示所有项**

SuperTag

InTouch SuperTag 可供定义复合型标记名。您可以定义包含多达 64 个成员标记名和 2 层嵌套的 SuperTag。成员标记名的行为与普通的标记名完全相同。它们支持趋势、报警以及所有的标记名**点域**。

如需有关 SuperTag 的详细信息，请参阅 “创建 InTouch SuperTag”。

扩展的标记名支持

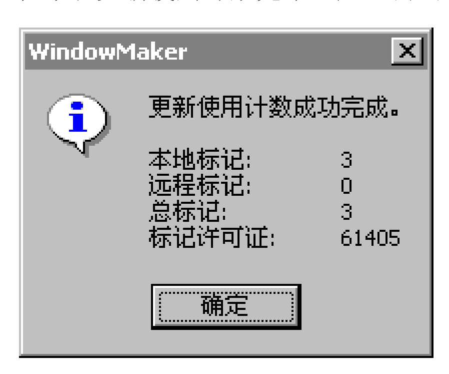
InTouch 在它的 “标记名字典”中能够支持多达 61405 个标记名。系统支持的标记名数目由软件许可证确定。

要确定系统的标记名支持功能

1. 关闭所有的窗口。
2. 在**特别**菜单上，单击**更新使用计数**。

   提示 有一个消息框会显示出来，告诉您更新使用计数可能要持续一段时间。此时，您可以取消命令或继续执行。

3. 单击**是**继续更新使用计数。
4. 在系统更新使用计数完毕之后，会出现下列对话框。



5. **标记许可证**行会显示许可证支持的标记名数。

备注 在 SPCPro 中用作 “采集标记”、“指示器标记”或 “触发器标记”的任何标记名均不计算在内。

6. 单击**确定**

定义新的标记名

标记名最长可达 32 个字符，但必须以希腊字符开头 （**A-Z** 或 **a-z**）。其余的字符可以是 A-Z、a-z、0-9、!、@、-、?、#、$、%、_、\ 以及 &。

备注 反斜杠 (\) 只对于 SuperTag 才有效。

标记名也是自动编号的。例如，如果输入并保存标记名 R4001，然后单击**新建**，则该标记名会自动编号为 R4002。如果标记名包含由一个字符分隔的数字，则会根据 InTouch 找到的第一个整数给它自动编号。例如，N7-0 将编号为 N7-1。注意这里只允许发生正向变化。例如， R4002 到 R4003、 R4003 到 R4004，等等。

在标记中使用连字符 (-) 时必须小心。它们在标记名中使用时是有效的，但它们在表达式或逻辑运算时也被用作负号或减号 “运算符”。因此，可能会引起混淆。

例如，如果在表达式中使用 `A=B-C`，您的意思是 `A = B` 减 `C` ，还是将名称为 **B-C** 的标记名赋值给名称为 **A** 的标记名呢？ InTouch 会假设是后者。为了防止出现此种情形，您可以在标记名和运算符之间插入空格。例如， `A = B - C`。

考虑这个例子： `X-101=FT-101*SP-101`

因为这里没有空格，是否可以看出， **FT-101** 将乘以 **SP-101**，然后再将值赋给 **X-101** ？

如需有关 “标记浏览器”的详细信息，请参阅 “标记浏览器”。

标记名字典对话框顶部的选项用于显示各种标记名详细资料级别的对话框：

对话框	描述
主要	显示主标记名字典对话框。在 SuperTag 的情况下，**主要**只显示父或根标记名。对父或根标记名所作的任何改变均将改写 “成员”标记名信息。在进行改变之后，请单击**保存**。此时会显示一个消息框，询问是否要使用根标记名的改变信息去改写成员标记名。
详细资料	显示所选标记名类型相应的详细资料级别对话框。
报警	显示所选标记名类型相应的报警配置对话框。

对话框	描述
详细资料和报警	显示所选标记名类型相应的详细资料和报警配置对话框。
成员	显示 SuperTag 型标记名的成员详细资料对话框。

提示 如果在任何“标记名字典”对话框中右击任意文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

选择标记

第一次访问“标记名字典”时，会显示内部系统标记名 **$AccessLevel** 的定义。在“标记名字典”中定义标记名之后，当您再次访问它时，则会显示最近编辑的标记名的定义。

单击 << 或 >> 可浏览“标记名字典”中当前存储的标记名定义。（没有可供显示的上一个或下一个标记名时，浏览按钮会被禁用）。

单击**选择**以迅速查找特定的标记名定义。此时会出现一个**选择标记**对话框，该对话框处于选择模式下。

要定义一个新标记名

1. 在**特别**菜单上，单击**标记名字典**，或者在“应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。

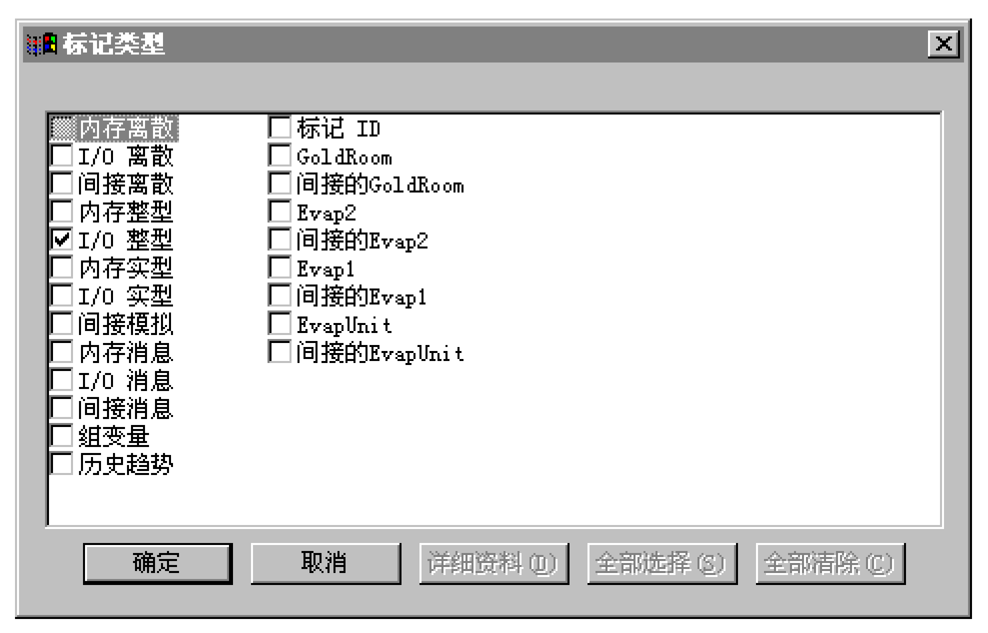


2. 单击**新建**。（此时**标记名**框的内容会被清除）。
3. 在**标记名**框中，输入要给新标记名使用的名称。

提示 标记名最长可达 32 个字符，但必须以希腊字符开头（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z、a-z、0-9、!、@、-、?、#、$、%、_、\ 以及 &。

您不能将 **RetVal** 这个词用作标记名。这是一个保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，则您会收到一则错误消息“变量不得使用该名称。标记已存在”。

4. 单击**类型**。此时会出现**标记类型**对话框。



5. 选择标记名类型，然后单击**确定**。此时会出现所选类型对应的详细资料对话框。（详细资料对话框将在本章稍后部分作介绍）。

提示 在 TemplateMaker 中创建的任何 SuperTag 的名称也将会出现在此对话框中，并可被选作标记类型。例如，上面所讲的 ColdRoom 和 EvapUnit。如果 SuperTag 不是使用 TemplateMaker 创建的，则会出现名称 "SuperTag"。例如，如果 SuperTag 是在动画链接标记名或表达式输入框、 QuickScript 或在外部文件中创建的，然后再加载到 DBLoad 实用程序。

如需有关标记名类型的详细信息，请参阅 “标记名类型”。

如需有关创建 InTouch SuperTag 的详细信息，请参阅“创建 InTouch SuperTag”。

备注 如果标记名当前链接到对象或用在 QuickScript 中，则只有当 WindowViewer 未在运行的情况下才能改变它的类型。

6. 单击**组**将标记名指定给特定的 “报警组”。此时会出现**报警组**对话框。选择要指定给该标记名的 “报警组”，然后单击**完成**。

备注 如果未将标记名指定给某个特定的 “报警组”，则缺省条件下， InTouch 会将它指定给根组 **$System**。

创建标记名并将它指定给某个 “报警组”之后，如果不关闭该对话框，则所有后来定义的标记名均会被指定给同一个 “报警组”，除非您改变它。

如需有关定义 “报警组”的详细信息，请参阅 第 9 章 “报警 / 事件”。

7. 对于 I/O 型标记名，选择**只读**以将标记名限制为在运行时处于只读状态。

8. 对于 I/O 型标记名，选择**读写**让标记名在运行时可以进行读、写。

9. 在**注释**框中，输入要系统储存的有关标记名的任何其它注释 （最多 50 个字符）。

提示 第一次访问**标记名字典**对话框时，**注释**框会显示内部系统标记名 **$AccessLevel** 的缺省注释。您应该删除此注释，以防它与您定义的任何标记名相关联。要删除注释，请选择它然后按 DEL 键。

如需有关 “分布式报警系统”的详细信息，请参阅第 9 章 “报警 / 事件”。

10. 如果希望在运行时只要工程单位值的变化超过指定的**记录死区**值时，标记名便会被记录到历史记录文件中，请选择**记录数据**；否则，在缺省条件下，不管值的变化如何，均会每小时记录一次。

备注 为了实际记录标记名，必须通过**特别**菜单上的**配置历史记录**命令，来启用历史记录功能。

如果后来您确定要清除此选项以便不再记录标记名，则会无法访问先前为该标记名记录的数据。此外，如果 WindowMaker 运行时您在 WindowMaker 中作了改变，则所作的改变在 WindowMaker 重新启动之前不会生效。

---

11. 如果希望记录由操作员、I/O、 QuickScript 或系统对标记名所作的所有数据值改变，请选择**记录事件**。

提示 定义标记名来监控事件时，在每次标记名的值发生改变的情况下，事件消息均会记入报警系统。事件消息会记录数据值是如何改变的。例如，是由操作员、 I/O、 QuickScript 还是由系统导致的改变。

选择**记录事件**时，将会激活**优先级**字段。您为**优先级**输入值确定了该标记名的事件优先级。此字段中的有效值是 1 到 999， 1 是最高优先级，而 999 是最低优先级。

如需有关事件和属性的详细信息，请参阅第 9 章 "报警 / 事件"。

12. 选择**保留值**，则每次 WindowViewer 退出时，都会保留标记名的当前值。每次 WindowViewer 重新启动时，此值会被用作标记名的初值。

备注 WindowViewer 重新启动时，保留值不会被写入 I/O 设备。 I/O 服务器第一次扫描设备时，会更新 I/O 值。

提示 WindowViewer 正在运行时，将不能选择或清除新的或现有标记名的保留值。选择此选项时，标记名的初始值会不断更新 , 以反映标记名的当前值。在 WindowViewer 退出时，初始值会根据最后一次保留的值进行设置。如果随后清除了这个选项，则标记名的初始值将设置为最后一次保留的值。

13. 如果要保留操作员对标记名的任何报警限字段值所作的更改，请选择**保留参数**。此值将会用作 WindowViewer 重新启动时报警的初始值。

备注 由于会立即记录改变，我们强烈推荐您只对不经常改变的值选择上述两个保留选项。

14. 定义标记名类型的详细资料，然后单击**关闭**。

定义标记名详细资料

最初显示的**标记名字典**对话框用于输入基本的标记名信息。许多采样点，尤其是输入和输出，需要更多的详细资料以便进行适当处理。对于指定的每种标记名，均会存在一个特定的详细资料对话框，供您指定该种标记名的详细资料。

大多数的标记名类型均有自己特定的详细资料对话框和报警条件对话框。缺省条件下，在选择标记名类型时，会出现它相应的详细资料对话框。

在完成基本标记名的定义之后，您将需要定义标记名和报警状态（如有必要）的详细资料。下面几节向您介绍定义每种标记名类型的详细资料时必须遵循的操作步骤。

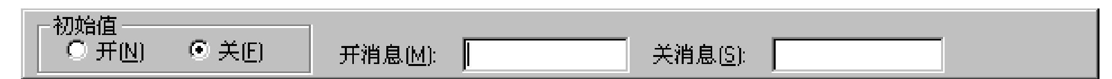
定义内存离散型标记名详细资料

内存离散型标记名存在于 InTouch 应用程序内部。需要一个值为 0 (False, Off) 和 1 (True, On) 的内部标记名时，您可以定义一个**内存离散**型标记名。

要定义内存离散型标记名详细资料

1. 选择**内存离散型**作为标记名类型时，会出现下面详细资料对话框。

提示 如果未出现，请单击**标记名**字典对话框顶部的**详细资料**。



2. 单击要在第一次加载运行时数据库时储存到该标记名中的**初始值**。
3. 如果要为此标记名定义一个离散报警状态，在标记名的值等于 1 (On, True) 时，其状态为 "on" （开），请在**开消息**框中输入要在报警窗口的数值 / 极限域中显示的消息。
4. 如果为此标记名定义一个离散报警状态，在该标记名的值等于 0 (Off, False) 时，其状态为 "on" （开），请在**关消息**框中输入要在报警窗口的数值 / 极限域中显示的消息。
5. 如果要为该标记名定义报警条件，请单击**标记名字典**对话框顶部的**报警**或**详细和报警**，以显示正在定义的标记名类型对应的报警条件对话框。

   如需有关报警的详细信息，请参阅第 9 章 “报警 / 事件”。
6. 在标记名定义完毕之后，单击**关闭**以保存标记名定义并关闭标记名对话框。

定义内存模拟型标记名详细资料

内存模拟型标记名存在于 InTouch 应用程序内部。内存模拟型标记名有两种：**内存整型**和**内存实型**。需要一个介于 -2,147,483,648 和 2,147,483,647 之间的 32 位有符号整数的内部标记名时，您可以定义一个**内存整型**标记名。

需要一个浮点值介于 $-3.4e^{38}$ 和 $3.4e^{38}$ 之间的内部标记名时，您可以定义一个**内存实型**标记名。（所有的浮点运算均按 64 位精度执行，但是会用 32 位来保存结果）。

要定义内存模拟型标记名详细资料

1. 选择**内存整型**或**内存实型**作为标记名类型时，会出现下面的详细资料对话框。

提示 如果未出现，请单击**标记名字典**对话框顶部的**详细资料**。

初始值(I):	0	最小值(N):	0	死区(D):	0
工程单位(E):		最大值(X):	9999	记录死区(B):	0

2. 在**初始值**框中，输入要在第一次加载运行时数据库时储存到该标记名中的值。
3. 在**最小值**框中，输入该标记名的最小值。这会用于 “历史趋势” 图、 I/O 及 **.Min EU** 标记名点域。
4. 在**最大值**框中，输入该标记名的最大值。这会用于 “历史趋势” 图、 I/O 及 **.Max EU** 标记名点域。
5. 在**工程单位**框中，输入要用于标记名的工程单位的标签。
6. 在**死区**框中，输入在更新数据库之前标记名的工程单位必须改变的量。
7. 在**记录死区**框中，输入标记名记入历史记录文件前标记名的工程单位必须改变的量。缺省值零表示会去记录每次改变。

备注 如果要在标记名的工程单位改变超过**记录死区**值时，将该标记名记录到磁盘中，则必须为该标记名选择**记录数据**。

如果在 WindowViewer 运行时改变**记录死区**值，则在停止并重新启动历史记录功能之前，您所作的改变不会生效。

如需有关历史记录的详细信息，请参阅第 12 章 “实时趋势和历史趋势”。

8. 如果要为该标记名定义报警条件，请单击**标记名字典**对话框顶部的**报警**或**详细和报警**，以显示所定义的标记名类型对应的报警条件对话框。

   如需有关报警条件的详细信息，请参阅 “定义标记名报警条件”。

9. 在标记名定义完毕之后，单击**关闭**以保存标记名定义并关闭标记名对话框。

定义内存消息型标记名详细资料

内存消息型标记名存在于 InTouch 应用程序内部。需要一个长达 131 个字符的内部文本字符串标记名时，您可以定义一个**内存消息**型标记名。

要定义内存消息型标记名详细资料

1. 选择**内存消息**作为标记名类型时，会出现下面的详细资料对话框。

   提示 如果未出现，请单击**标记名字典**对话框顶部的**详细资料**。

最大长度(L): 131 初始值(I):

2. 在**最大长度**框中，输入允许在该标记名的消息使用的最大字符数。（InTouch 最多允许使用 131 个字符，并将其显示为缺省值）。
3. 在**初始值**框中，输入 WindowViewer 最初启动时标记名所显示的文本字符串。
4. 在标记名定义完毕之后，单击**关闭**以保存标记名定义并关闭标记名对话框。

定义 I/O 离散型标记名详细资料

所有从另一个 Windows 程序读取值或将值写入另一个 Windows 程序的标记名均为 I/O 型标记名。这包括可编程控制器、过程计算机、其它 Windows 程序的所有输入和输出，以及来自网络节点的数据。

需要一个值为 0 (False, Off) 或 1 (True, On) 的 I/O 型标记名时，您可以定义一个 **I/O 离散**型标记名。

要定义 I/O 离散型标记名详细资料

1. 选择 **I/O 离散型**作为标记名类型，会出现下面的详细资料对话框。

   提示 如果未出现，请单击**标记名字典**对话框顶部的**详细资料**。



2. 单击要在第一次加载运行时数据库时储存到该标记名中的**初始值**。(**Off** 等于 **0**， **On** 等于 **1**）。此值不会写入 I/O 设备。

3. 单击要在更新运行时数据库时应用于该值的**输入转换**：

输入转换	描述
直接	I/O 输入值从服务器程序直接读取而不加改变。
取反	从服务器程序读取时，反转 I/O 输入值。例如，如果 I/O 输入值在服务器程序中是 0，则 InTouch 将自动反转、存储它并将其显示为 1。

4. 如果要为此标记名定义一个离散报警状态，在标记名的值等于 1 (On, True) 时，其状态为 "on" （开），请在**开消息**框中输入要在报警窗口的数值 / 极限域中显示的消息。

5. 如果为此标记名定义一个离散报警状态，在该标记名的值等于 0 (Off, False) 时，其状态为 "on" （开），请在**关消息**框中输入要在报警窗口的数值 / 极限域中显示的消息。

6. 单击**访问名**，以定义或选择要赋给此标记名的 “访问名”。（如果此按钮的右边已经显示 “访问名”，并且您没有定义或选择一个不同的访问名，则它会被赋给该标记名）。

   如需有关 “访问名” 的详细信息，请参阅第 13 章 “I/O 通讯”。

7. 在**项目**框中，为标记名将从中读 / 写其值的服务器程序中的数据点输入有效的项目名。例如，如果要从一个 PLC 寄存器读取数值，请为该寄存器输入一个有效的标识作为项目名。

   提示 项目名会自动索引编号。例如，如果输入并存储项目名 R4001，然后单击**新建** （以定义一个新的标记名），则该项目名将自动编号为 R4002。如果项目名包含一个字符分隔的数字，则 InTouch 会根据找到的第一个整数给它自动编号。例如， N7-0 将编号为 N7-1。注意这里只允许发生正向变化。例如， R4002 到 R4003、 R4003 到 R4004，等等。

8. 如果要将该标记名用作项目名，请选择**使用标记名作为项目名**选项。

9. 如果要为该标记名定义报警条件，请单击**标记名字典**对话框顶部的**报警**或**详细和报警**，以显示所定义的标记名类型对应的报警条件对话框。

   如需有关报警条件的详细信息，请参阅 “定义标记名报警条件”。

10. 在标记名定义完毕之后，单击**关闭**以保存标记名定义并关闭标记名对话框。

定义 I/O 模拟型标记名详细资料

所有从另一个 Windows 程序读取值或将值写入另一个 Windows 程序的标记名均为 I/O 型标记名。这包括可编程控制器、过程计算机、其它 Windows 程序的所有输入和输出，以及来自网络节点的数据。内存模拟型标记名有两种：**I/O 整形**和 **I/O 实型**。

需要一个包含介于 -2,147,483,648 和 2,147,483,647 之间的 32 位有符号整数的 I/O 标记名时，您可以定义一个 **I/O 整型**标记名。

需要一个浮点值介于 $-3.4e^{38}$ 和 $3.4e^{38}$ 之间的 I/O 标记名时，您可以定义一个 **I/O 实数**型标记名。（所有的浮点运算均按 64 位精度执行，但是会用 32 位来保存结果）。

要定义 I/O 离散型标记名详细资料

1. 选择 **I/O 整型** 或 **I/O 实型**作为标记名类型时，会出现下面的详细资料对话框。

   提示 如果未出现，请单击**标记名字典**对话框顶部的**详细资料**。



2. 在**初始值**框中，输入要在第一次加载运行时数据库时储存到该标记名中的值。此值不会写入 I/O 设备。
3. 在**死区**框中，输入在更新数据库之前标记名的工程单位可以改变的量。
4. 在**最小工程单位**框中，输入在收到最小的原始计数值时标记名的工程单位值。
5. 在**最小原始数据**框中，输入原始 I/O 整型值低嵌位的最小值。
6. 在**最大工程单位**框中，输入在收到最大的原始计数值时标记名的工程单位值。
7. 在**最大原始数据**框中，输入原始 I/O 整型值高嵌位的最大值。

提示 您可以使用**最小工程单位**、**最小原始数据**、**最大工程单位**和**最大原始数据**值定标 I/O 标记名。

如需有关给标记名定标的详细信息，请参阅 “定标 I/O 标记名”。

8. 在**工程单位**框中，输入要给标记名的工程单位使用的标签。
9. 选择**转换**类型，在按如下方式计算工程单位时，数据库会使用此转换来定标原始计数值：

   如果选择**线性**，则使用端点之间的线性插值法计算结果。输入值的线形换算法则是：
   ```
   EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) /
       (MaxRaw - MinRaw)) + MinEU
   ```
   输出值的线性换算法则是：
   ```
   RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) /
       (MaxEU - MinEU)) + MinRaw
   ```
   如果选择**平方根**，则原始计数值通常用于插值。这对于调整从非线性设备 （如压力传感器）输入的数据非常有用。输入值的平方根换算法则是：
   ```
   EUValue = sqrt(RawValue - MinRaw) * ((MaxEU - MinEU) /
       sqrt(MaxRaw - MinRaw)) + MinEU
   ```
   输出值的平方根换算法则是：
   ```
   RawValue = square((EUValue - MinEU) * (sqrt(MaxRaw –
       MinRaw) / (MaxEU -MinEU))) + MinRaw
   ```
10. 单击**访问名**，以定义或选择要赋给此标记名的 “访问名”。（如果此按钮的右边已经显示 “访问名”，并且您没有定义或选择一个不同的访问名，则它会被赋给该标记名）。

    如需有关 “访问名” 的详细信息，请参阅第 13 章 “I/O 通讯”。
11. 在**项目**框中，为标记名将从中读 / 写其值的服务器程序中的数据点输入一个有效的项目名。例如，如果要从一个 PLC 寄存器中读取数值，请给该寄存器输入一个有效的标识作为项目名。

提示 项目名会自动索引编号。例如，如果输入并存储项目名 R4001，然后单击**新建** （以定义一个新的标记名），则该项目名将自动编号为 R4002。如果项目名包含一个字符分隔的数字，则 InTouch 会根据找到的第一个整数给它自动编号。例如， N7-0 将编号为 N7-1。注意这里只允许发生正向变化。例如， R4002 到 R4003、 R4003 到 R4004，等等。

12. 如果要将该标记名用作项目名，请选择**使用标记名作为项目名**选项。

13. 如果要为该标记名定义报警条件，请单击**标记名字典**对话框顶部的**报警**或**详细和报警**，以显示所定义的标记名类型对应的报警条件对话框。

    如需有关报警条件的详细信息，请参阅 “定义标记名报警条件”。

14. 在**记录死区**框中，输入在将标记名记入历史记录文件之前标记名的工程单位必须改变的量。

    备注 如果要在标记名的工程单位改变超过**记录死区**值时，将标记名记录到磁盘中，则必须为该标记名选择**记录数据**。

    如果在 WindowViewer 运行时改变**记录死区**值，则在停止并重新启动历史记录功能之前，您所作的改变不会生效。

    如需有关历史记录功能的详细信息，请参阅第 12 章 “实时趋势和历史趋势”。

15. 在标记名定义完毕之后，单击**关闭**以保存标记名定义并关闭标记名对话框。

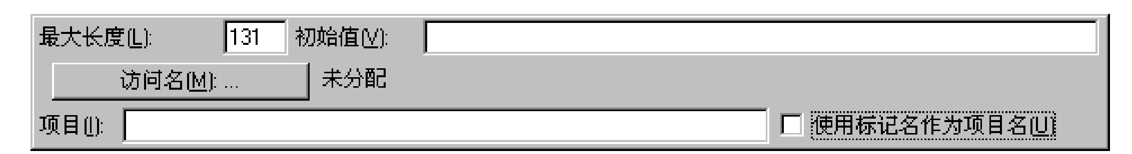
定义 I/O 消息型标记名详细资料

所有从另一个 Windows 程序读取值或将值写入另一个 Windows 程序的标记名均为 I/O 型标记名。这包括可编程控制器、过程计算机、其它 Windows 程序的所有输入和输出，以及来自网络节点的数据。需要从 I/O 设备采集文本字符串时，您可以定义一个 **I/O 消息**型标记名。 I/O 消息型标记名的长度限制在 131 个字符以内。

要定义 I/O 消息型标记名详细资料

1. 选择 **I/O 消息**作为标记名类型时，会出现下面的详细资料对话框。

   提示 如果未出现，请单击**标记名字典**对话框顶部的**详细资料**。



2. 在**最大长度**框中，输入允许在该标记名的消息使用的最大字符数。（InTouch 最多允许使用 131 个字符，并将其显示为缺省值）。

3. 在**初始值**框中，输入在 WindowViewer 最初启动时标记名所要显示的文本字符串。此值不会写入 I/O 设备。

4. 单击**访问名**，以定义或选择要赋给此标记名的 “访问名”。（如果此按钮的右边已经显示 “访问名”，并且您没有定义或选择一个不同的访问名，则它会被赋给该标记名）。

   如需有关 “访问名”的详细信息，请参阅第 13 章 “I/O 通讯”。

5. 在**项目**框中，为标记名将从中读 / 写其值的服务器程序中的数据点输入一个有效的项目名。例如，如果要从一个 PLC 寄存器读取数值，请为该寄存器输入一个有效的标识作为项目名。

   提示 项目名会自动索引编号。例如，如果输入并存储项目名 R4001，然后单击**新建** （以定义一个新的标记名），则该项目名将自动编号为 R4002。如果项目名包含一个字符分隔的数字，则 InTouch 会根据找到的第一个整数给它自动编号。例如， N7-0 将编号为 N7-1。注意这里只允许发生正向变化。例如， R4002 到 R4003、 R4003 到 R4004，等等。

6. 如果要将该标记名用作项目名，请选择**使用标记名作为项目名**选项。

7. 在标记名定义完毕之后，单击**关闭**以保存标记名定义并关闭标记名对话框。

定义 SuperTag 成员标记名详细资料

成员标记名是在 SuperTag 模板中定义的。成员标记名的行为和普通标记名完全相同，并且可以是 “离散”型、“整型”、“实型”、“消息”型或另一个 SuperTag。与普通的 InTouch 标记名一样，成员标记名支持趋势、报警和所有的标记名**点域**。

如需有关成员标记名的详细信息，请参阅 “创建 InTouch SuperTag”。

定义标记名并选择 SuperTag 模板作为其标记名类型时，缺省情况下，在 SuperTag 模板中定义的所有成员标记名均会被设置为 “内存”数据访问类型。如果这是您所需要的类型，则不必进行特殊配置。不过，如果您需要将 SuperTag 模板中的任何成员标记名定义为 I/O 类型，则必须进行一些额外的配置。

要定义 I/O SuperTag 成员标记名

1. 选择 SuperTag 模板作为标记名类型时，会出现下面的详细资料对话框。

提示 如果未出现，请单击**标记名字典**对话框顶部的**成员**。

成员列表: Beef\RoomTemp :: I/O 整型 数据访问: 内存 I/O

请注意，您在**标记名**框中输入的新标记名会变成**成员表**中所有成员标记名的“父辈”。

2. 单击**成员表**箭头，然后从列表中选择要定义为 I/O 数据访问类型的成员标记名。
3. 在**数据访问**组中，选择 **I/O**。此时会出现该成员标记名类型｛“离散”、“模拟”（实型或整型）或“消息”｝对应的 I/O 详细资料对话框。
4. 与对待普通的 InTouch I/O 型标记名一样，输入所需的 I/O 详细资料。
5. 要保存所作的更改，请选择列表中的另一个成员标记名并配置它，或单击**关闭**。

定义标记名报警条件

您可以在定义标记名的同时定义其报警条件。报警详细资料对话框共有两种类型。一种用于离散型标记名，另一种用于模拟（整型或实型）型标记名。

报警约束标记名

作为可选项，您可以给每种报警（或报警子状态）指定一个可禁止该报警的标记名。该标记名为 TRUE（非零或非空）时，该报警将被有效禁止。与此类似，在标记名为 FALSE（零或空）时，则不会禁止该报警。您可以在 WindowMaker 通过“标记名字典”来指定**报警约束标记**。**报警约束标记**不能在运行时改变。不过，在运行时可以改变**报警约束标记**的值。

报警可以被有效禁止、停用，或同时被有效禁止和停用。如果它当前被停用或有效禁止，则它将会“关闭”。这表示，如果您尝试启用当前被有效禁止的报警，则它将不会“打开”。

备注 **报警约束标记**是普通的标记名，它会被包括在使用计数与许可证限制中。

您也可以使用一些只读标记名点域来返回一个字符串，该字符串包含**报警约束标记**的名称。这些点域是：

- AlarmDscInhibitor
- AlarmLoLoInhibitor
- AlarmLoInhibitor
- AlarmHiHiInhibitor

- AlarmHiInhibitor
- AlarmMajDevInhibitor
- AlarmMinDevInhibitor
- AlarmRocInhibitor

这些点域会返回标记名的名称。因此，您可以在 InTouch QuickScript 中的间接型标记名引用中使用该名称，来找出**报警约束标记**的当前值，或改变**报警约束标记**的值。如此一来，您就可以在运行时强制启用或有效禁止报警组。

如需有关这些报警点域的详细信息，请参阅联机 *Intouch 参考指南*。

禁止报警分为两个阶段：

1. 指定约束标记名。
2. 将约束标记名的状态从 FALSE 更改为 TRUE，反之亦然。

与停用一样，对于包含子状态的报警，每个子状态均可单独禁用。每个子状态均可由不同的标记名来禁用。与停用一样，被禁用的报警 （此时该标记名为 TRUE）不会等待*确认*。如果报警包含子状态，则它只能等待*确认*仍可用的子状态。

只要转换导致报警脱离**有效禁止**状态，则会执行检查逻辑来确定 InTouch 是否应将该项目放入*报警*状态。

如果报警变为**有效禁止**，而项目仍处于*报警*状态，则该项目会被强制转换为一个不同的 （有效）状态。具体的状态应取决于当前何种状态可用，以及它们是否也被停用或有效禁止。此活动是由 InTouch 根据报警类型、极限值等进行处理的。

如果报警 （或报警子状态）变为**有效禁止**但仍等待*确认*，则该项目会被强制设置为一个不同的 （有效）状态。与项目是否应报警一样，InTouch 必须确定此状态应该是哪种。

如需有关报警的详细信息，请参阅第 9 章 “报警 / 事件”。

定义离散型标记名报警条件

离散报警对应于离散型标记名。您可以配置*报警*状态是对应于离散型标记名的 TRUE (On, Yes, 1) 状态还是 FALSE (Off, No, 0) 状态，以及与报警关联的*优先级*。

要定义离散型标记名的报警条件

1. 在**特别**菜单上，单击**标记名字典**，或者在 “应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。

2. 单击**标记名字典**对话框顶部的**报警**或**详细和报警**，以显示离散型报警详细资料对话框：



3. 选择要使用的**确认模型**选项：

   - **条件**：确认会统计至确认时为止所有进入报警状态或子状态的转换情况。（这是缺省值，其作用与先前的版本一样）。
   - **面向事件**：确认只针对进入报警状态或子状态的特定转换；只有在针对<u>最近一次</u>此类转换时，确认才会被接受。
   - **扩展的摘要**：确认只是针对特定转换进行的，无论是转入报警状态、子状态还是恢复正常状态。每次从正常状态的转换均标志着一个新 RTN 组的开始。 RTN 组中的所有转换均须单独予以确认，之后整个 RTN 组才被视为已确认。

备注 在使用**扩展的摘要** “确认模型”定义标记名时，不得将**报警属性**对话框中的**返回意味着确认**选项应用于该标记名。

任何一个单独的报警均可配置为使用这些确认模型中的任一个。

如需有关详细信息，请参阅第 7 章的 “报警确认模型”部分。

1. 在**报警注释**框中，输入要用于 **.AlarmComment** 点域的缺省注释（最多 131 个字符）。（这是一个可选点域）。

   如需有关详细信息，请参阅 “标记名报警注释”。

2. 单击报警时标记名所处的**报警状态**。
3. 在**优先级**框中，输入一个介于 1 和 999 之间的值 （1 是最高优先级，999 是最低优先级）。您可以将这个优先级值用于选择要显示在窗口中、记录到磁盘或打印出来的报警。
4. 要将约束标记名与报警条件关联，请单击该报警条件对应的**报警约束标记**省略号 (...) 按钮。此时会出现**选择标记**对话框。

备注 选定相应的报警条件时，每个**报警约束标记**省略号 (...) 按钮均会被激活。您不能在**报警约束标记**输入框中输入内容。

5. 在**选择标记**对话框中，双击要用于抑制报警条件的标记名。此时对话框将关闭，所选标记名会出现在**报警约束标记**框中。

   如需有关详细信息，请参阅 “报警约束标记名”。

6. 在完成所有的输入之后，单击**关闭**保存标记名定义并关闭所有的标记名对话框。

定义模拟型标记名的报警条件

模拟型报警对应于模拟型标记名。您可以将*报警*状态配置为是否对应于标记名的任意模拟值，以及与该报警关联的优先级。

要定义模拟型标记名的报警条件

1. 在**特别**菜单上，单击**标记名字典**，或者在 “应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。

2. 单击**标记名字典**对话框顶部的**报警**或**详细和报警**，以显示模拟型报警详细资料对话框：



3. 选择要使用的**确认模型**选项：

   - **条件**：确认会统计至确认时为止所有进入报警状态或子状态的转换情况。（这是缺省值，其作用与先前的版本一样）。
   - **面向事件**：确认只针对进入报警状态或子状态的特定转换；只有在针对<u>最近一次</u>此类转换时，确认才会被接受。
   - **扩展的摘要**：确认只是针对特定转换进行的，无论是转入报警状态、子状态还是恢复正常状态。每次从正常状态的转换均标志着一个新 RTN 组的开始。 RTN 组中的所有转换均须单独予以确认，之后整个 RTN 组才被视为已确认。

备注 使用**扩展的摘要**“确认模型”定义标记名时，不得将**报警属性**对话框中的**返回意味着确认**选项应用于该标记名。

任何一个单独的标记名均可配置为使用这些确认模型当中的任一个。所选的确认模型将应用于该标记名上所有类型的报警。

如需有关详细信息，请参阅第 7 章的“报警确认模型”部分。

4. 在**报警注释**框中，输入要用于 **.AlarmComment** 点域的缺省注释（最多 131 个字符）。

   如需有关详细信息，请参阅“标记名报警注释”。

5. 选择在模拟型标记名的值超过绝对极限时用于检测的报警类型（**LoLo**、 **Low**、 **High**、 **HiHi**）。

6. 在**报警值**框中，为该报警输入极限值。

   例如，在 **LoLo** 和 **Low** 报警的情形下，只要标记名的值小于**报警值**，则会存在报警条件。在 **High** 和 **HiHi** 报警的情形下，只要标记名的值大于**报警值**，就会存在报警条件。这些点域支持使用实数（如 100.75）。

7. 在**值死区**框中，输入必须降到报警值之下多少个工程单位才会消除报警。

   例如，要让报警条件恢复到正常状态，则标记名的值不仅必须返回报警限之内，而且返回量必须超过指定的**值死区**。**值死区**可防止由于报警的不断反复引起的“过度”报警（即标记名值在报警限附近上下徘徊、连续的进入和退出报警状态）。

8. 选择在模拟型标记名的值为指定**目标**值的主或副偏差时要用于检测的偏差（**副**和**主偏差**）报警类型。

9. 在**偏差百分比**框中，输入模拟型标记名可以偏离**目标**值以产生副或主偏差报警条件的百分比。它是用标记名范围百分比来表示的。在标记名详细资料对话框中输入的“最小工程单位”和“最大工程单位”值定义了该范围。

10. 在**目标**框中，输入标记名的所需值或参考值，副偏差和（或）主偏差百分比正是以此为基础。

    例如，我们假设以下整型标记名的设置：

    最小值 = -1000

    最大值 = 1000

    副偏差百分比 = 10

    主偏差百分比 = 15

    目标值 = 500

    在标记名的总范围是 **-1000 到 +1000 或 2000** 时，要计算“副偏差”或“主偏差”报警值，请将 **2000** 乘以“副偏差”或“主偏差”百分比：（**2000 x .10 （副） = 200**）。如果“目标值”是 **500**，则每当标记名的值低于 **300** 或高于 700 时，就会产生“副偏差”报警。

11. 在**偏差死区百分比**框中，输入在报警消除之前标记名值必须降到极限值以下的偏差百分比。

12. 如果要检测指定时间间隔内报警值的变化是否超出范围，请选择**变化率**。每当标记名的值发生改变时，系统都会测试此标记名并确定是否应发出**变化率**报警。此时，变化率是使用上次的值、上次更新时间、当前值以及当前时间进行计算的。计算所得的值会与报警条件所允许的变化率进行比较。如果变化率大于报警限，则会给该标记名设置**变化率**报警条件。**变化率**报警将持续生效，直至在下一个时间间隔内标记名值的变化小于该变化范围。

13. 在 **% /** 框中，输入允许的最大变化百分比。

14. 为变化的时间间隔单位选择**秒**、**分**或**小时**。

15. 在相应的**优先级**框中，输入一个介于 1 和 999 的数字（1 是最高优先级， 999 是最低优先级）。您可以使用此优先级值来选择要显示在窗口中、记录到磁盘或打印出来的报警。

16. 要将**报警约束标记**与某个报警条件相关联，请单击报警条件相应的**报警约束标记**省略号 (...) 按钮。此时会出现**选择标记**对话框。

    备注 选定相应的报警条件时，每个**报警约束标记**省略号 (...) 按钮均会被激活。您不能在**报警约束标记**输入框中输入内容。

17. 在**选择标记**对话框中，双击要用于抑制报警条件的标记名。此时对话框将关闭，所选标记名会出现在**报警约束标记**框中。

    如需有关详细信息，请参阅“报警约束标记名”。

18. 在完成所有的输入之后，单击**关闭**保存标记名定义并关闭所有的标记名对话框。

标记名报警注释

在过去， InTouch 允许您进行配置，以便在产生报警时将标记名注释附加到该报警。这个标记名注释可以显示在报警向导或报警日志中。InTouch 还提供了一种配置设置，供操作员在确认报警时输入注释。该确认注释将更新 “标记名数据字典”中的标记名注释。

为配合 “分布式报警系统”的其它配置选项，新版本扩充了 “标记名数据字典”中每个标记名的信息存储量。现在，您可以创建专门用于报警的全新注释域、输入更长的内容，并将它与标记名注释完全分开。

报警注释是一个新的字符串型点域，可通过 InTouch QuickScript 进行设置或读取。该点域的名称是 **.AlarmComment**。作为一个 InTouch 字符串，它可以容纳多达 131 个字符。在 “报警注释”对话框中，用户可以为这个点域输入缺省设置。

作为新版本的一项功能改进，报警不再使用标记名注释作为报警注释。相反，现在使用的是 **.AlarmComment** 点域。此外，如果操作员在确认报警时输入一个新注释，并且应用程序被配置为将此注释复制到 “标记名数据字典”，则 **.AlarmComment** 会更新 – 而不是标记名注释。

由于这些改进都涉及对 “标记名数据字典”的更改，因此文件 tagname.x.x 的版本号已经更新为新的修订级。因此，如果 Window Maker 在加载包含先前版本 “标记名数据字典”的应用程序，则它会自动将旧版 “标记名数据字典”转换为当前的版本。作为此项转换的一部分，标记名注释会作为缺省设置被复制到报警注释中。

每个报警确认均附有注释 – 无论该确认是通过 “分布式报警对象”、脚本函数还是任何其它方式进行的。确认报警的操作员可以使用此注释来添加有关报警的信息。

在报警变为活动状态时，“分布式报警系统”会创建一个 “报警记录”来跟踪该报警实例。对于与报警起因有关的注释， InTouch 使用在 “标记名字典”中输入的 “报警注释”。如果操作员在确认报警时提供注释，则 InTouch 会将它作为该报警实例的 “确认注释”添加到 “报警记录”中。报警起因与确认的注释均保存在 “分布式报警系统”中，并且二者均记入报警数据库。“分布式报警对象”和 “报警打印程序”会根据报警实例是否已获得确认来分别显示 “报警注释”或 “确认注释”。下次同一个标记上出现报警时，该 “报警注释”将再次用于新报警实例的起因，操作员可以在确认新实例时输入不同的 “确认注释”。

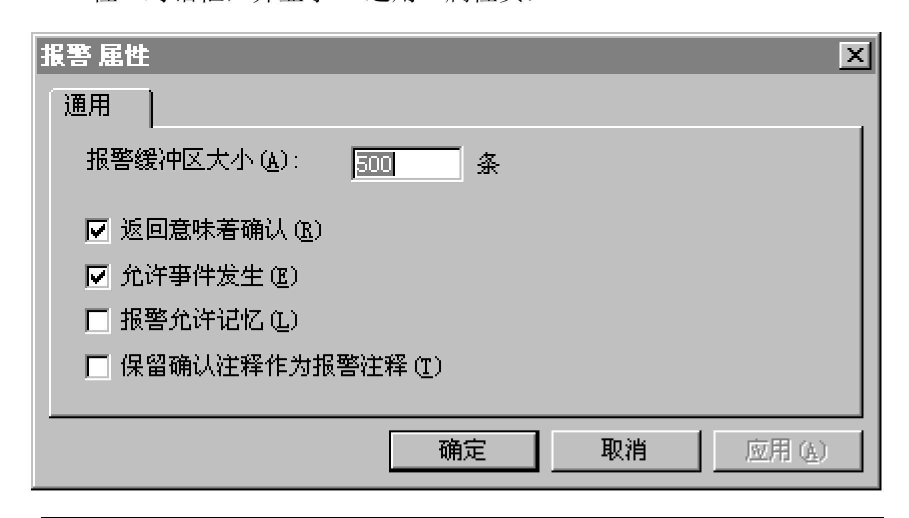
您也可以选择使用 “确认注释”来更新标记名数据库中的 “报警注释”。如果您启用此项功能，则 AlarmComment 点域会在运行时被改写，包括 “标记名数据字典”中的 “报警注释”项。

要将运行时发生的改变写入标记名数据库中的**报警注释**点域，请为当前应用程序在 INTOUCH.INI 文件 （位于存储当前应用程序的目录）中添加下面一行代码：

```
CommentRetentive=1
```

您也可以在 WindowMaker GUI 中配置此项功能，具体如下：

1. 在**特别**菜单上，指向**配置**，然后选择**报警**。此时会出现 “报警属性”对话框，并显示 “通用”属性页：



提示 如果右击任何一个报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 如果需要保留确认报警时输入的注释作为对相应标记名 AlarmComment 点域的更新，并将它复制到 “标记名字典”中，请选择**保留确认注释**。如果未选中此复选框，则 “确认注释”仍会同已确认的报警一起显示 （在数据库、打印输出和显示画面中），但 “报警注释”将保持不变。

3. 单击 **OK**。

创建 InTouch SuperTag

InTouch 支持一种模板结构，可供定义 SuperTag 这种复合型标记名。SuperTag 模板最多可容纳 64 个成员标记名和 2 个嵌套级别。这表示，SuperTag 父项可包含多达 64 个内嵌的子成员，并且这每个子成员又可以包含多达 64 个子成员标记名，因此总共可以有 4095 个成员标记名。（当一个 SuperTag 模板父项内嵌到另一个 SuperTag 模板时，它就变成了一个 “子成员”）。所有 SuperTag 模板子成员标记名的行为与普通的标记名完全相同。它们支持趋势、报警以及所有的标记名**点域**。

为方便起见， InTouch 提供了一个 "TemplateMaker" 供您用来创建 SuperTag。 TemplateMaker 允许您创建、编辑和删除 SuperTag 模板及其成员标记名。 InTouch 将所有的 SuperTag 模板均保存到 InTouch 安装目录 （不是应用程序目录）下的 supertag.dat 文件。这样您创建的模板便可以用作任何应用程序中。

InTouch 还提供了其它几种方法供您创建 SuperTag。例如，您可以从“标记名字典”、动画链接标记名或表达式输入框、 InTouch QuickScript 中创建 SuperTag，此外，您还可以在外部文件中进行创建，然后使用 InTouch DBLoad 实用程序将它加载到应用程序中。

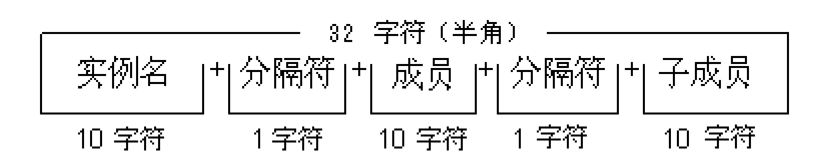
如需有关详细信息，请参阅 “创建 SuperTag 的其它方法”。

在创建 SuperTag 父模板时，它的名称会自动添加到 “标记名字典”中的标记名**标记类型**对话框中，并可立即供您在创建新标记名时选用。您不必重新启动 WindowMaker，就可以定义标记名来使用新创建的 SuperTag 类型。

注意！ 如果修改现有的 SuperTag 模板，则该 SuperTag 的全部现有的实例将不会受到影响。（实例指的是在 “标记名字典”中定义的、将该 SuperTag 用作类型的标记名）。换句话说，您对 SuperTag 所作的修改不可回溯。改变，使用修改后的 SuperTag 定义的所有新实例将使用新的结构。与此类似，如果通过其它方法给 SuperTag 实例添加一个成员标记名，则它的模板也不会更新。

InTouch SuperTag 语法

由于 InTouch 标记名限制在 32 字符内，每个 SuperTag 的父辈实例 \ 子辈成员 \ 子成员均限制为最多使用 32 个字符。 SuperTag 引用的层数最多只能为两个模板 （父辈实例 \ 子辈成员）和一个成员，具体如下所示：



SuperTag 模板中的每个成员均可使用现在用于访问普通 InTouch 标记名类型的**点域**的标准格式去访问。 InTouch 中凡是可以使用普通标记名的地方，都支持使用 SuperTag 引用语法。例如，有效的 SuperTag 引用可以是：

```
ColdRoom4\EvapUnit1\FanMotor2.MaxEU
```

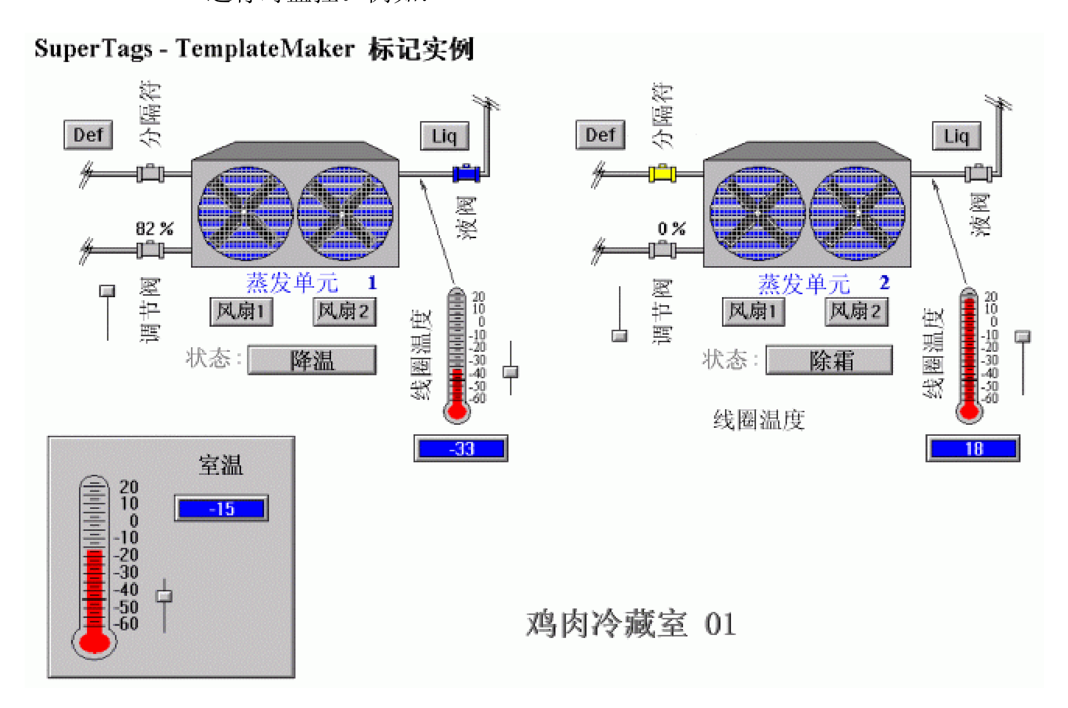
远程标记名引用也支持 SuperTag。语法示例：

```
PLC1:"Turkey\EvapUnit2\PrsRegVlv.EngUnits"
```

如需有关使用远程标记名引用的详细信息，请参阅 “远程标记名引用”。

创建 SuperTag 模板结构

为了形象化地描述工厂环境中的 SuperTag 概念，我们假定有四个同样的冷藏室，其中储存着牛肉、猪肉、鸡肉和火鸡。每个冷藏室都有一个室内温度，以及两台脱水装置。每台脱水器装置有七个数据值我们需要在运行时监控。例如：



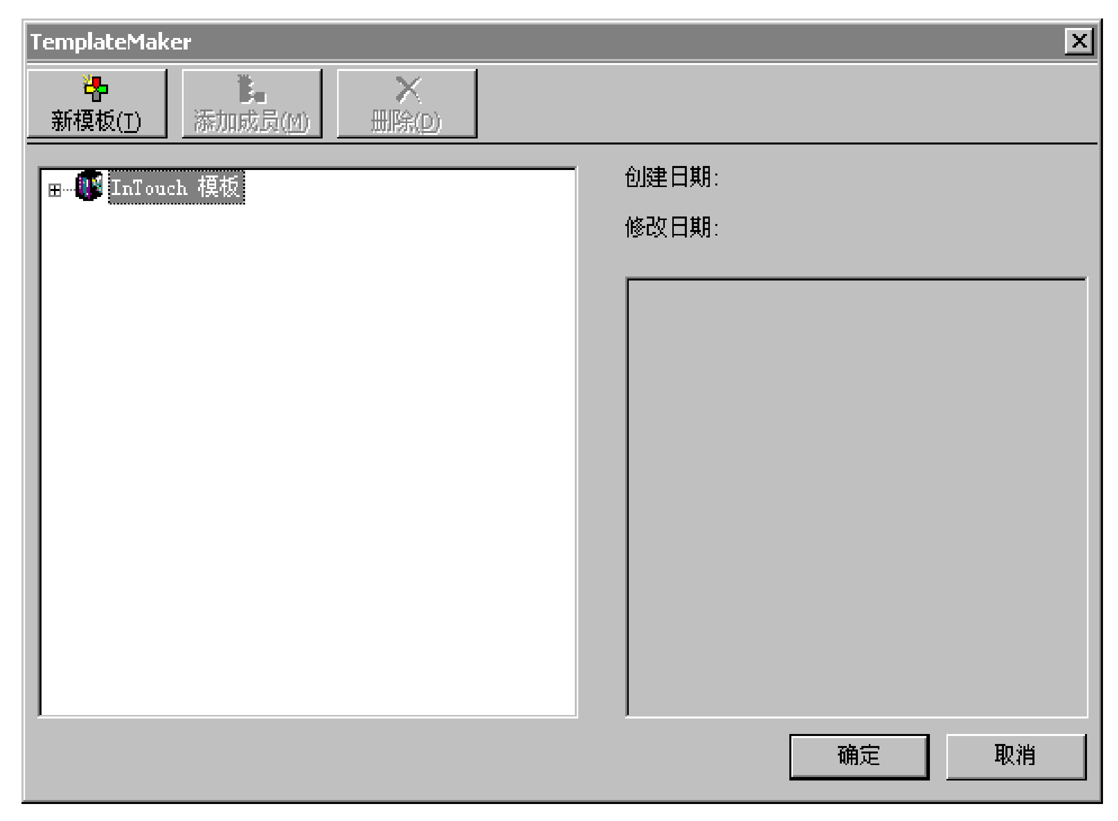
如果我们不创建 SuperTag 模板来实现这个目的，就需要为每个冷藏室中的每个数据值手工定义一个单独的标记名，然后乘以冷藏室的总数。换句话说，我们必须在 “标记名字典” 中组织管理并定义数十个标记名！

通过使用 SuperTag，我们就可以节约数小时的开发时间并最大限度降低出错机会。对于上述 “冷藏室” 的情形，我们将创建一个 SuperTag 父模板 "EvapUnit"。（这个 "EvapUnit" 稍后将变成 ColdRoom 父模板的子成员。这是一种 “逐渐细化” 的设计概念）。 EvapUnit 可以使用七个子成员标记名进行定义：

成员标记	类型	描述
FanMotor1	离散型	风扇 1 的马达启动器
FanMotor2	离散型	风扇 2 的马达启动器
DefrostVlv	离散型	除霜气体阀状态
LiquidVlv	离散型	液体冷却阀状态
CoilTemp	实型	致冷剂的温度
PrsRegVlv	整型	压力调节阀 (0-100%)
EvapStatus	消息型	脱水装置状态字符串

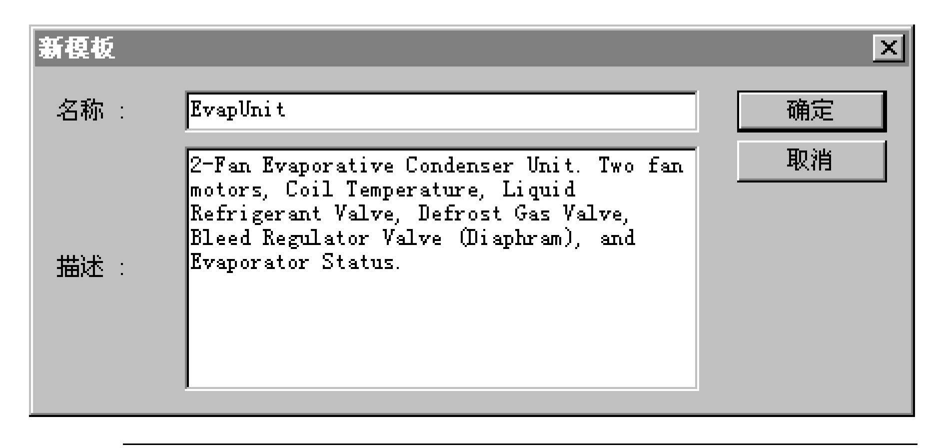
要创建 **SuperTag** 父模板

1. 在**特别**菜单上，单击 **TemplateMaker**，或者在 “应用程序浏览器” 中，双击 **TemplateMaker**。此时会出现 **TemplateMaker** 对话框。



2. 在 TemplateMaker 窗口中，选择 **InTouch 模板**，然后单击**新模板**，或者右击 **InTouch 模板**，然后选择**新模板**。此时会出现**新模板**对话框。

提示 您也可以选择 **InTouch 模板**，然后右击窗口的空白区。

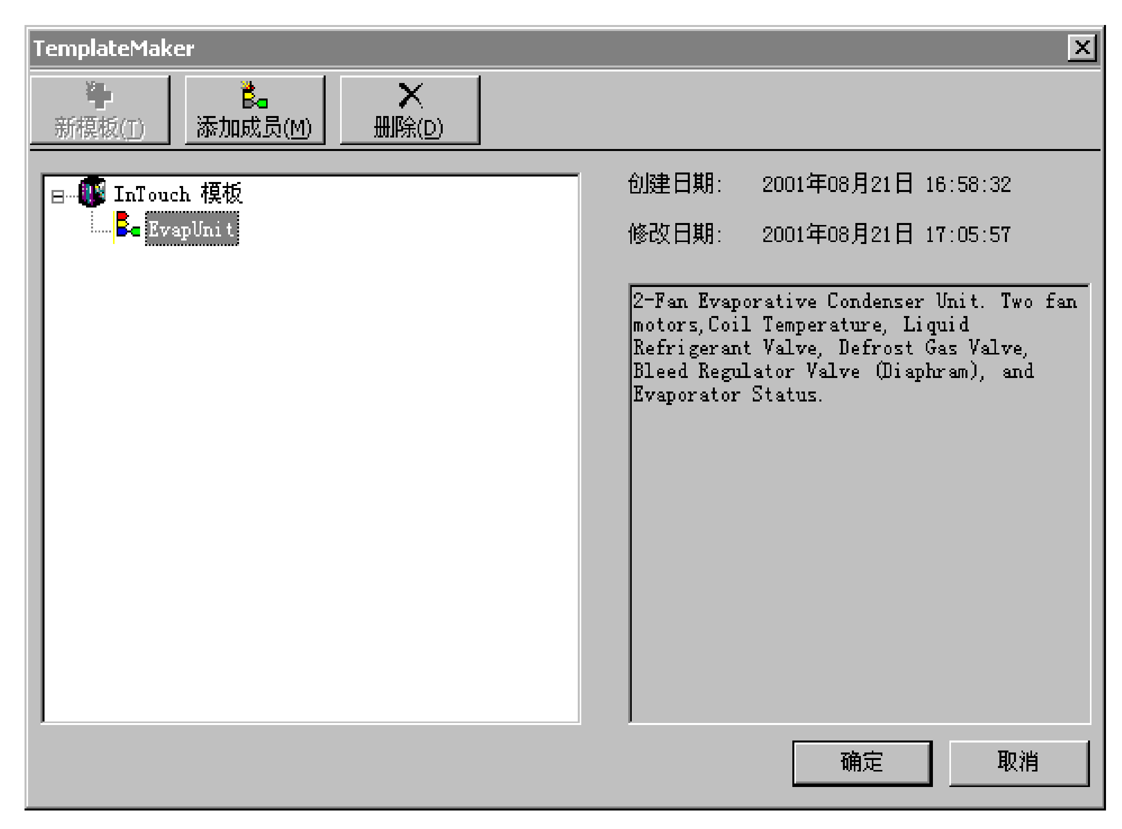


提示 如果您右击任何 TemplateMaker 对话框中的文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。

3. 在**名称**字段中，为新模板输入一个唯一的名称（最多 10 个字符）。

> **提示** 添加新的父模板时，它们的名称会作为标记名类型立即显示在"标记名字典"的**标记名类型**对话框中，并可立即供您选用。您不必重新启动 WindowMaker，就可以定义新的标记名并将它们赋给 SuperTag 类型。

4. 在**描述**字段中，输入用于描述该模板的任何信息。
5. 单击**确定**。此时 **TemplateMaker** 对话框会再次出现，在它的窗口中显示新的模板名。

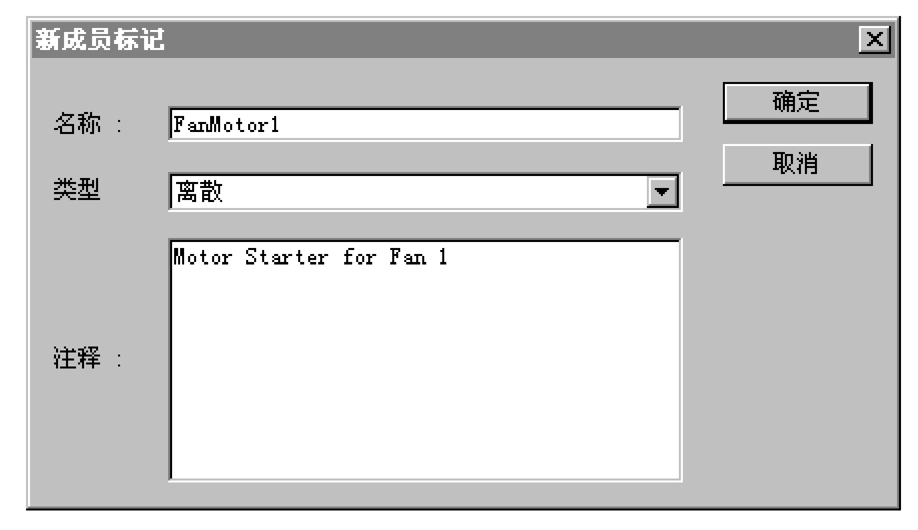


创建模板之后，**新成员**和**删除**按钮即被激活。选择模板名时，该模板的创建日期和时间和（或）最近一次修改日期和时间、以及模板描述也会显示出来。

备注 TemplateMaker 窗口以分层列表的方式显示当前定义的所有 SuperTag 父模板及其子成员。要展开模板视图，请使用鼠标左键单击模板名称旁边的⊞ 。为父模板名定义的所有成员标记名都会显示出来。要折叠视图，请使用鼠标左键单击⊟。

要创建 SuperTag 成员标记名

1. 在 TemplateMaker 窗口中，选择 SuperTag 模板 （在本例中是 EvapUnit），然后单击**添加成员**，或者右击 SuperTag 模板名，然后单击**添加成员**。此时会出现**新成员标记**对话框。



2. 在**名称**框中，输入要给成员标记名使用的名称。
3. 在**类型**框中，输入该成员的标记名类型，然后单击**类型**箭头，并从列表中选择一种类型。类型可以是 “离散”型、“整型”、“实型”、“消息”型或另一个 SuperTag 模板。

提示 如果输入类型的第一个字母，则列表框中以该字母开头的第一个类型会自动显示在框中。如果有多个类型以该字母开头，则您可以继续输入名称的下一个字母，以此类推。

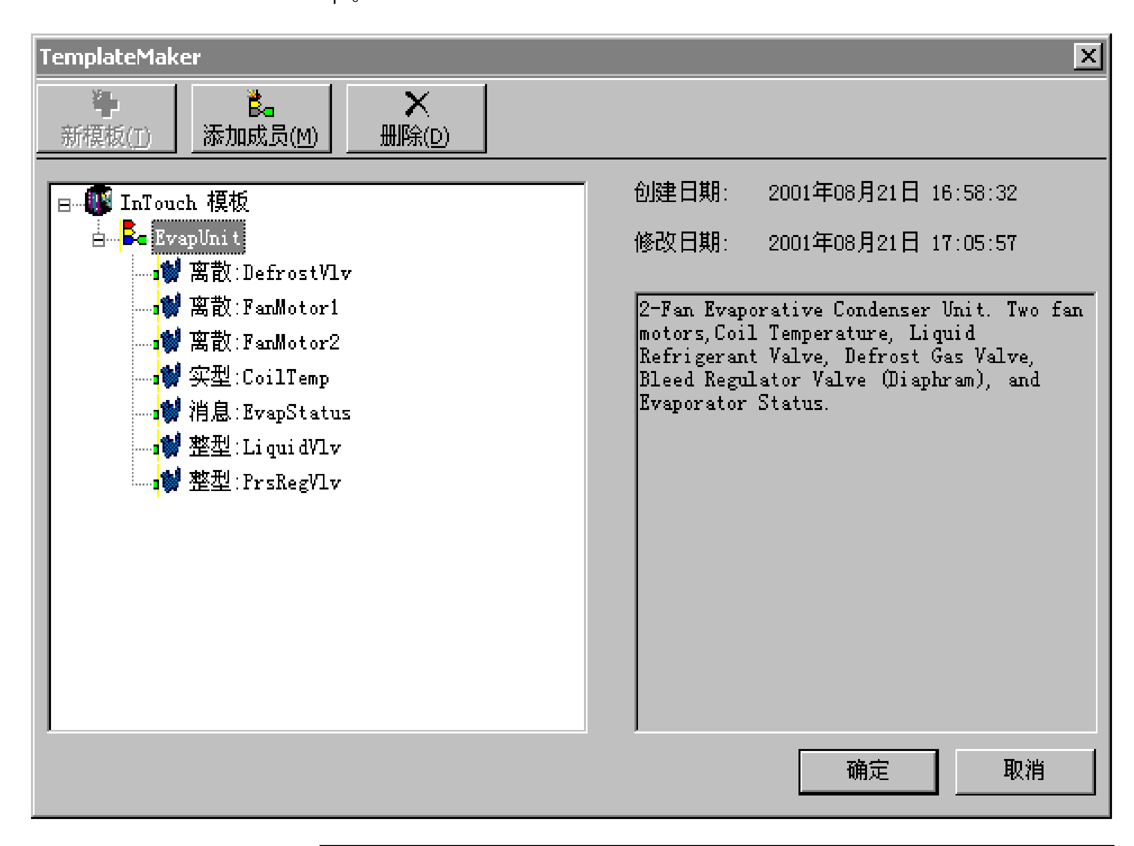
备注 此处指定的类型仅仅是 SuperTag 模板的占位符。缺省条件下，在 TemplateMaker 中进行定义时，所有成员标记名均设置为 “内存”型。当您在 “标记名字典”中定义模板实例时，则需要指定这些标记名实际上是 “内存”型还是 “I/O”型标记名。

如需有关**成员**对话框的详细信息，请参阅 “定义 SuperTag 成员标记名详细资料”。

4. 在**注释**字段中，输入用于描述成员标记名的任何信息。
5. 单击**确定**。

提示 重复此过程，以便将其它成员标记名添加到 SuperTag 模板。

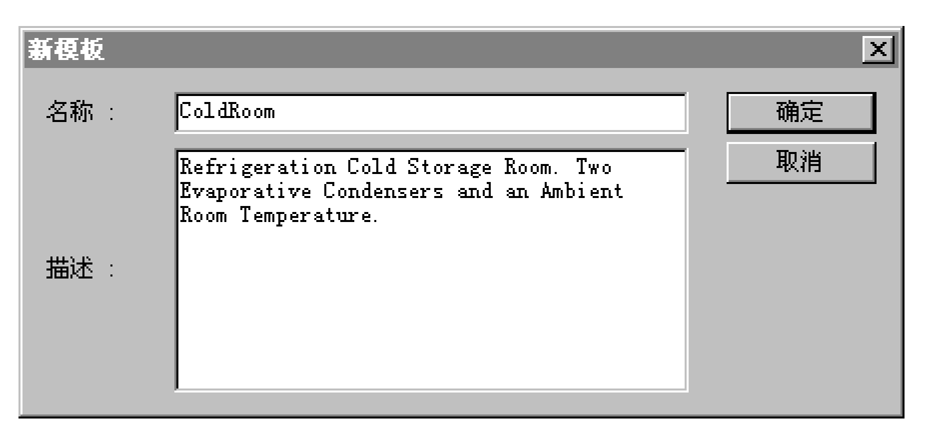
新的成员标记名会添加到 TemplateMaker 窗口中的 SuperTag 父模板下。



提示 请注意，如果选择某个成员标记名，则**新成员**按钮将失效，因为成员只能在现有的 SuperTag 父模板下创建。选择成员时，该成员标记名的创建日期和时间、最近一次修改日期和时间，以及模板描述均会显示出来。

6. 现在，您将要创建另一个父模板 ColdRoom。 ColdRoom 将包含一个成员标记名 RoomTemp 和两个 EvapUnit 子成员模板 （EvapUnit1 和 EvapUnit2）。这两个子成员模板将使用父 SuperTag 模板 EvapUnit 作为它们的类型。

7. 在 TemplateMaker 窗口中，选择 **InTouch 模板**，然后单击**新模板**，或者右击 **InTouch 模板**，然后单击**新模板**。此时会出现**新模板**对话框。



8. 在**名称**字段中，为新的父模板输入一个唯一的名称 （最多 10 个字符）。

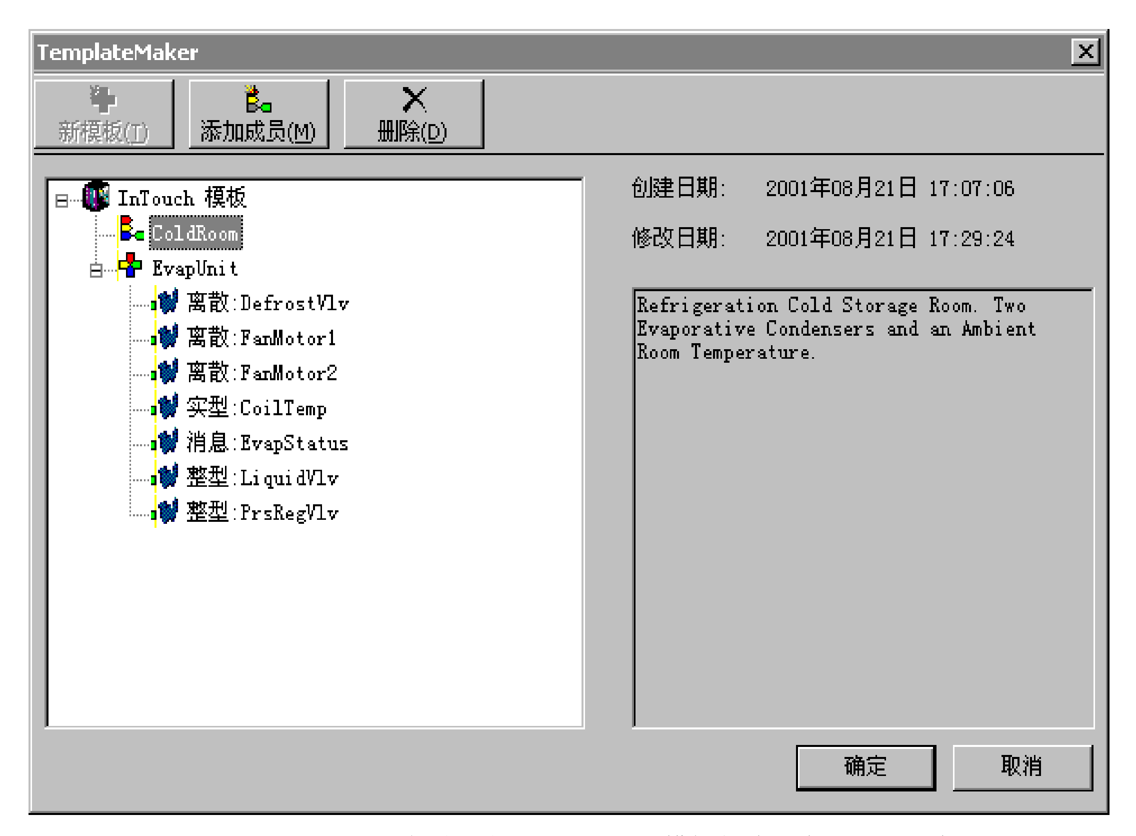
提示 添加新的父模板时，它们的名称会作为标记名类型立即显示在 "标记名字典"的**标记名类型**对话框中，并可立即供您选用。您不必重新启动 WindowMaker，就可以定义新的标记名并将它们赋给 SuperTag 类型。

---

9. 在**描述**字段中，输入用于描述模板的任何信息。
10. 单击**确定**。此时会再次出现 **TemplateMaker** 对话框，在它的窗口中显示新的模板名：

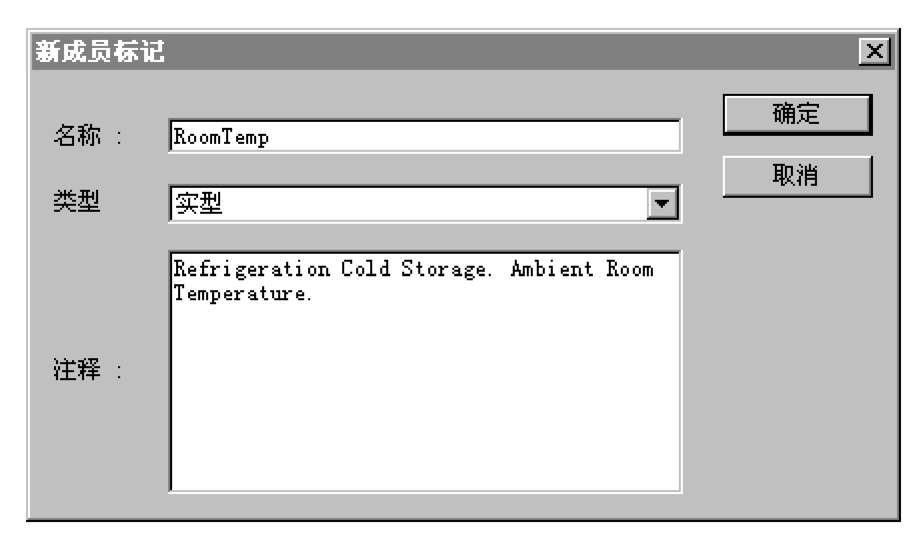
11. 单击**确定**。

    父模板将添加到 TemplateMaker 窗口的 **InTouch 模板**列表中。



12. 我们现在需要为 ColdRoom 父模板创建三个成员；两个 EvapUnit 子成员和一个名为 RoomTemp 的成员标记名。

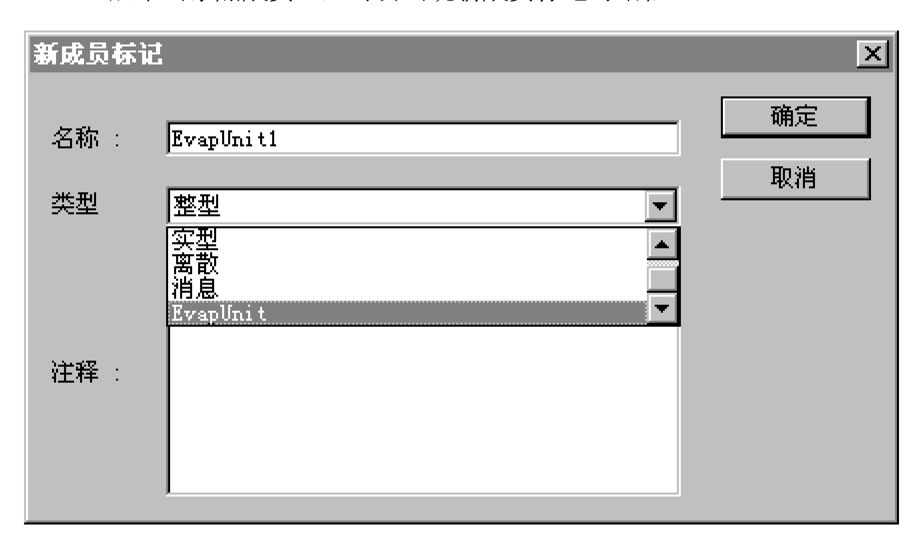
13. 在 TemplateMaker 窗口中，选择 SuperTag 父模板（在本例中是 ColdRoom），然后单击**添加成员**，或者右击 SuperTag 父模板名，然后单击**添加成员**。此时会出现**新成员标记**对话框。



14. 在**名称**框中，输入要给成员标记名使用的名称。
15. 在**类型**框中，输入该成员的标记名类型，然后单击**类型**箭头，并从列表中选择一种类型。类型可以是“离散”型、“整型”、“实型”、“消息”型或另一个 SuperTag 模板。
16. 在**注释**字段中，输入用于描述成员标记名的任何信息。
17. 单击**确定**。

接着，我们将创建两个子成员模板：EvapUnit1 和 EvapUnit2，它们使用 EvapUnit 模板类型。

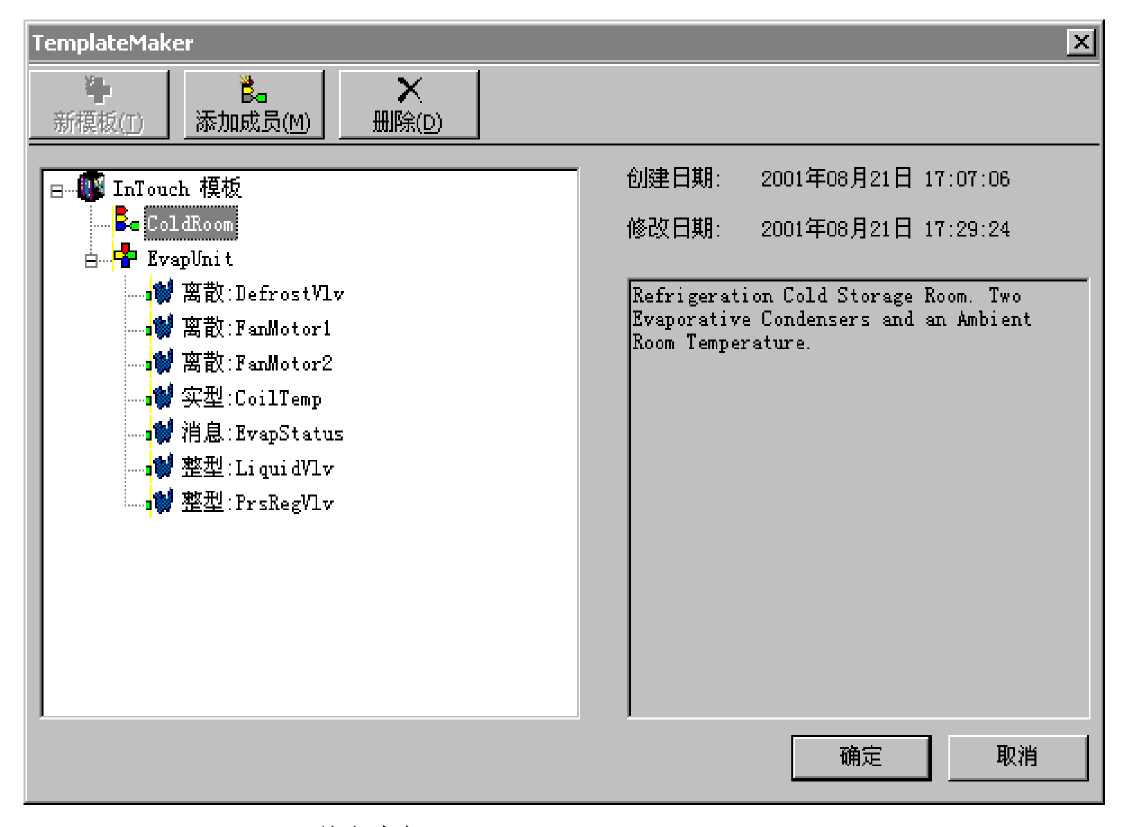
1. 在 TemplateMaker 窗口中，选择 SuperTag 父模板 （在本例中是 ColdRoom），然后单击**添加成员**，或者右击 SuperTag 父模板名，然后单击**添加成员**。此时会出现**新成员标记**对话框。



2. 在**名称**框中，输入要给成员标记名使用的名称。
3. 在**类型**框中，输入该成员的标记名类型，然后单击**类型**箭头，并从列表中选择一种类型。在本例中，我们可以使用特殊的模板类型 EvapUnit。
4. 在**注释**字段中，输入用于描述成员标记名的任何信息。
5. 单击**确定**。

提示 对 EvapUnit2 重复上述过程。

在我们已经完成 ColdRoom 父模板之后，TemplateMaker 窗口将显示下列模板结构层次。



单击**确定**。

至此我们已完成 ColdRoom SuperTag 模板，现在我们可以立即创建将该模板用作标记名类型的标记名实例。

如需有关详细信息，请参阅 “定义 SuperTag 模板实例”。

创建间接 SuperTag

在定义新的 SuperTag 标记类型时， InTouch 会自动定义对应的间接 SuperTag 标记类型。这个对应的间接 SuperTag 完全是一个副本，它包括原件的所有成员标记名。



副本中所有成员的标记类型均会自动设置为间接。例如，假设您定义了下列 SuperTag 标记名类型：

MySuperTagType

离散型	MyDiscrete
整型	MyInteger
实型	MyReal
消息型	MyMessage

InTouch 使用下列结构定义间接 SuperTag 标记名类型：

Indirect MySuperTagType

间接	离散型	MyDiscrete
间接	模拟型	MyInteger
间接	模拟型	MyReal
间接	消息型	MyMessage

在脚本中，间接 SuperTag 标记类型的行为类似于 “间接离散型”、“间接模拟型” 或 “间接消息型” 等标记类型。例如，请考虑下列语句：

```
MyIndirectSuperTag.Name = "MySuperTag";
```

如果等号右边的名称指 SuperTag 父项， InTouch 会遍历其子项，并尝试指定一个值给左边间接 SuperTag 的对等成员。如果 InTouch 不能找到对等的成员，则 InTouch 将不会给该特定的子项赋值。

假定上述 SuperTag 具有下列结构：

MyIndirectSuperTag	MySuperTag
Value1	Value1
Value2	Value2
Value3	Value3
ValueX	Value4
	Value5

上面的语句具有下列效果：

```
MyIndirectSuperTag.Value1 = MySuperTag.Value1
MyIndirectSuperTag.Value2 = MySuperTag.Value2
MyIndirectSuperTag.Value3 = MySuperTag.Value3
```

此外，间接 SuperTag 标记类型的作用与动画链接中的 “间接离散型”、“间接模拟型” 或 “间接消息型” 等标记类型相同。例如，“按钮动作” 脚本可以执行下列语句：

```
MyIndirectSuperTag.Name = "MySuperTag";
```

随后，“模拟值显示” 链接可以使用下列表达式来显示十倍 MySuperTag.Value1 的值：

```
MyIndirectSuperTag.Value1 * 10
```

编辑 SuperTag 模板和成员标记名

您可以随时修改 SuperTag 模板或成员标记名。不过，如果修改现有的 SuperTag 模板或其成员，则该模板现有的全部实例不会受到影响。（实例指的是在 “标记名字典” 中定义的、将该 SuperTag 用作类型的标记名）。换句话说，您对 SuperTag 所作的修改不可回溯。不过，所有使用修改后的 SuperTag 定义的新实例将使用新的结构。

要编辑现有的 SuperTag 模板或成员标记名

1. 在 TemplateMaker 窗口中，双击 SuperTag 模板名 （或成员名），或右击它然后单击**编辑**。此时会出现**编辑模板** （或**编辑成员标记**）对话框，显示 SuperTag 模板 （或成员）的定义。
2. 执行所需的编辑，然后单击**确定**。

要删除 SuperTag 模板或成员

1. 在 TemplateMaker 窗口中，选择要删除的 SuperTag 模板名 （或成员名），或右击它然后单击**删除**。此时会出现一个消息框，要求您确认删除。
2. 单击**是**删除所选名称，或单击**否**取消删除。

> **备注** 如果按下 ESC 键而不是**确定**来关闭 TemplateMaker，则该模板不会被删除。删除模板时，与它关联的所有成员标记名也会被删除。

定义 SuperTag 模板实例

在 TemplateMaker 中一个很重要的概念是区分 SuperTag 模板和模板实例。模板实例是 SuperTag 模板的一个特定的具体示例。模板与实例之间最重要区别是父模板名会被实例标记名取代。子模板名和子成员标记名<u>不会改变</u>。

例如，这相当于一个用来设计蜡纸之类的文字模板，该蜡纸用来制作实际的图画。图画自身，打个比方来说，就是 “模板实例”，它们按照模板或蜡纸的样子进行创建。

再回到 ColdRoom 模板的情形，在我们创建模板之后，从其中我们可以创建 SuperTag 实例 “牛肉”、“猪肉”、鸡肉” 和 “火鸡”。为此，我们只要简单地创建四个将 ColdRoom 用作其类型的标记名。这样我们只要做一次，就可以在 “标记名字典” 中快速创建 60 个标记名。从而节约了大量的时间！

在创建 "ColdRoom" SuperTag 模板和实例后，我们可以通过使用有效的 SuperTag 引用，在动画链接表达式或 QuickScript 中引用它的任何成员。例如：

```
Beef\RoomTemp

Chicken\RoomTemp.RawValue

Chicken\EvapUnit1\FanMotor1.OnMsg

Pork\EvapUnit2\EvapStatus

Turkey\EvapUnit2\PrsRegVlv.EngUnits
```

如需有关定义模板实例的详细信息，请参阅 “定义 SuperTag 成员标记名详细资料”。

要使用新创建的模板来创建 SuperTag

1. 在**特别**菜单上，单击**标记名字典**，或者在“应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。



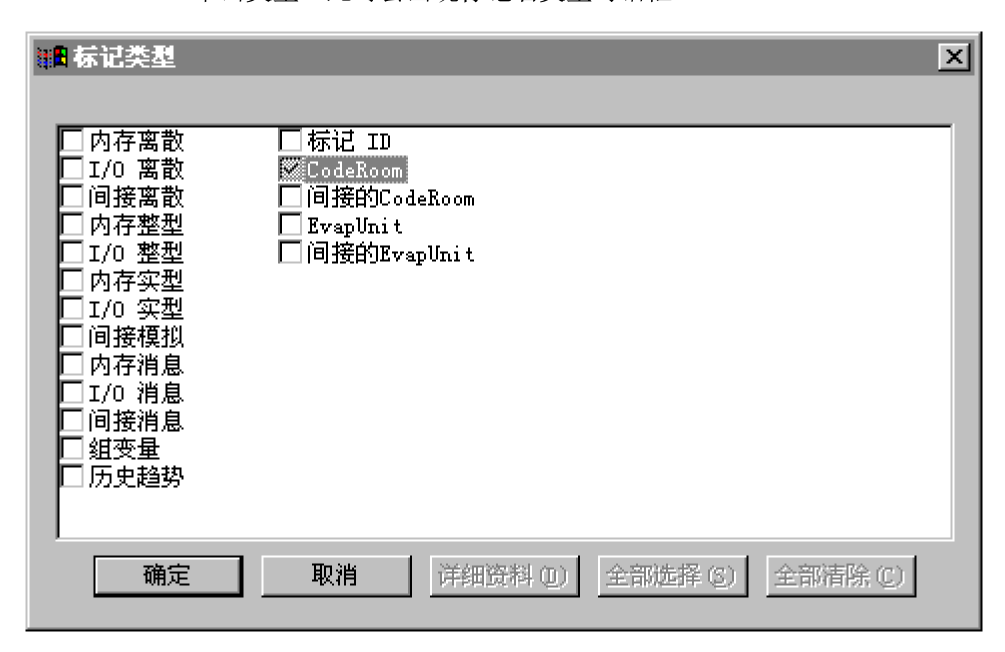
2. 单击**新建**。（此时**标记名**框的内容会被清除）。
3. 在**标记名**框中，输入要给新标记名使用的名称。

提示 标记名最长可达 32 个字符，但必须以希腊字符开头（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z、a-z、0-9、!、@、-、?、#、$、%、_、\ 以及 &。

您不能将 **RetVal** 这个词用作标记名。这是一个保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，则您会收到一则错误消息“变量不得使用该名称。标记已存在”。

---

4. 单击**类型**。此时会出现**标记名类型**对话框。



5. 为标记名选择 **ColdRoom**，然后单击**确定**。

6. 选择 SuperTag 模板作为标记名的类型时，会出现下面的详细资料对话框。

提示 如果未出现，请单击**标记名字典**对话框顶部的**成员**。

成员列表: Beef\RoomTemp :: I/O 整型 数据访问: 内存 I/O

请注意，您在**标记名**框中输入的新标记名会变成**成员表**中所有成员标记名的 “父辈”。

7. 单击**成员表**箭头，然后从列表中选择要定义为 I/O 数据访问类型的成员标记名。
8. 在**数据访问**组中，选择 **I/O**。此时会出现该成员标记名类型 {“离散型”、“模拟型” (实型或整型) 或 “消息型”} 对应的 I/O 详细资料对话框。
9. 与对待普通的 InTouch I/O 型标记名一样，输入所需的 I/O 详细资料。
10. 要保存所作的更改，请选择列表中的另一个成员标记名并配置它，或单击**关闭**。

提示 如需有关标记名类型的详细信息，请参阅 “标记名类型”。

如需有关创建 InTouch SuperTag 的详细信息，请参阅 “创建 InTouch SuperTag”。

要复制一个从模板创建的 SuperTag 实例

1. 在标记名字典中选择 SuperTag，然后单击**新建**。
2. 此时会出现一个对话框，询问是否要复制。如果单击**是**，在会出现一个对话框，提示输入新 SuperTag 的名称。输入 10 个字符长的名称，然后单击**确定**。

备注 此功能不适合通过动画链接表达式、 InTouch QuickScript，以及通过 DBLoad 实用程序加载到 “标记名字典”中的外部 .csv （逗号分隔变量）文件等方式创建的 SuperTag。

至此您已按照原件创建了一个新的 SuperTag 实例。唯一的区别是根或父项名称。

创建 SuperTag 的其它方法

除了 TemplateMaker 之外， InTouch 还支持通过动画链接表达式、 InTouch QuickScript，以及通过 DBLoad 实用程序加载到 “标记名字典”中的外部 .csv （逗号分隔变量）文件等方式来创建 SuperTag。不过，您也可以通过 “标记名字典”将成员或子成员添加到现有的 SuperTag 中，这是最简便的一种方法。

如需有关使用 DBLoad 创建 SuperTag 的详细信息，请参阅 “创建 SuperTag 实例”。

备注 在使用其它方法创建成员时，该成员不能反映到 TemplateMaker 中的 SuperTag 模板定义里。

在通过动画链接表达式或 InTouch QuickScript 创建 SuperTag 时，必须使用有效的 SuperTag 格式。例如：



下面的语法示例是有效的：

```
ParentInstance\ChildMember
   ParentInstance\ChildMember\Submember
```

下面的语法示例是无效的：

```
ParentInstance\
```

```
ParentInstance\ChildMember\
```

如果使用的格式无效，则出现一个错误消息框，提醒您该语法无效。

提示 如果在动画表达式或 QuickScript 中指定的 SuperTag 实例和成员标记名目前尚未定义，则会出现一个消息框，询问是否要立即去定义它。单击**确定**。此时会出现 “标记名字典”，显示指定的 SuperTag 实例和成员标记名。

使用标记名字典创建 SuperTag

“标记名字典” 是创建 SuperTag 实例或成员标记名的最简单的替代方法。

要在标记名字典中创建 SuperTag

1. 在**特别**菜单上，单击**标记名字典**，或者在“应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。单击**新建**。



2. 在**标记名**框中，输入 SuperTag 实例的准确名称，后跟反斜线 (\) 分隔符和新成员标记名的名称。此本例中，我们输入 **Turkey\RoomTemp2**。

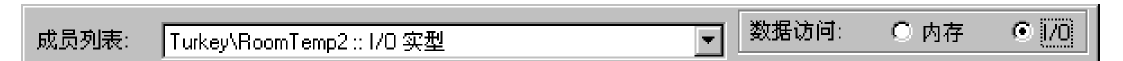
备注 给现有的 SuperTag 实例添加新成员标记名时，实例名的拼写必须完全相符。否则，会添加一个全新的 SuperTag 实例和成员。

提示 标记名最长可达 32 个字符，但必须以希腊字符开头（**A-Z** 或 **a-z**）。其余的字符可以是 A-Z、a-z、0-9、!、@、-、?、#、$、%、_、\ 以及 &。

您不能将 **RetVal** 这个词用作标记名。这是一个保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，则您会收到一则错误消息“变量不得使用该名称。标记已存在”。

---

3. 单击**类型**，并选择 SuperTag 成员的类型。（在这里不需要去选择其它的选项）。此本例中，我们选择了“I/O 实型”。
4. 单击**保存**或**关闭**以添加该成员。
5. 要在不退出“标记名字典”的情况下查看 Turkey SuperTag 中的成员标记名，请单击左或右双向箭头按钮。此时会出现**成员**详细资料对话框：



6. 单击**关闭**以关闭“标记名字典”。
7. 如果在 SuperTag 显示于“标记名字典”时单击**新建**，则会出现下面的对话框，询问是否需要制作与显示的 SuperTag 实例完全相同的副本：

备注 给现有的 SuperTag 实例添加新成员标记名时，实例名的拼写必须完全相符。否则，会添加一个全新的 SuperTag 实例和成员。



8. 单击**是**以创建另一个 SuperTag 实例，它与显示的 SuperTag 实例完全相同。此时会出现**输入名称**对话框。

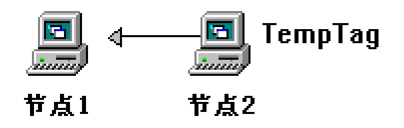


9. 输入新的 SuperTag 实例名。
10. 单击**确定**。

提示 “标记名字典”会为新的 SuperTag 实例自动创建所有的成员标记名和子成员标记名，它们可立即用于动画链接和 InTouch QuickScript。

远程标记名引用

InTouch 为工厂自动化应用提供了真正的客户端 / 服务器架构。不必使用本地“标记名字典”中的任何标记名，就可以设计出客户端应用程序。这可以通过使用 InTouch 的“远程标记名引用”功能来实现。例如：



在本示例中，您可以用两种方法获取 Node2 上标记名 "TempTag" 的值：

1. 在 Node1 的“标记名字典”中创建一个 I/O 型标记名，在与该 I/O 标记名关联的“访问名”中将 Node2 用作**节点名**。

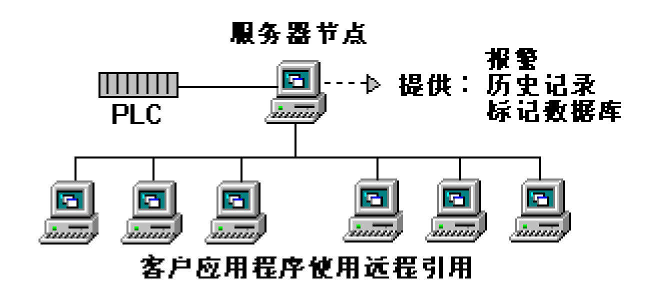
2. 使用一个指向 "TempTag" 的直接远程引用。例如，
   PLC1:"TempTag"

换句话说，在窗口或 QuickScript 中，您既可以引用本地标记名，也可以使用 *AccessName:"item"* 来引用远程标记名。

如需有关远程标记名引用语法的详细信息，请参阅 “远程标记名引用语法”。

要直接引用任何其它 FactorySuite 应用程序中的远程标记名时，只需要 *AccessName:"item"* 就可以了。您不必在本地 “标记名字典” 中定义该远程标记名。远程引用也可以访问来自任何 I/O 数据源 （如 Wonderware I/O Server 或 Microsoft Excel）的数据。此外，它们还支持 SuperTag。远程标记名引用 SuperTag 的有效语法是：*Accessname:"ParentInstance\ChildMember\SubMember"*。

此外，在使用远程标记名引用并导入窗口或 QuickScript 时，所有要做的事情就是将占位符标记名转换为远程标记名引用。您不必在本地 “标记名字典” 中定义标记名。远程引用可以从网络上的任何 FactorySuite 应用程序进行访问，具体如下图所示：



远程标记名引用语法

远程标记名引用的有效语法是 *AccessName:"item"*。远程引用中使用的字符与标记名的有效字符相同。有效的字符是：A-Z、 a-z、 0-9、 !、 @、 -、 ?、 #、 $、 %、 _、 \ 以及 &。如果您确定在远程标记名引用中没有使用任何无效字符，则不必用引号将 *item* 部分括起。

提示 远程标记名引用中的 "item" 部分也可以使用标记名**点域**。例如，"MyAlarm.HiHi"。

为了使用任何其它字符，您必须将 *"item"* 部分用引号括起。例如，如果使用 ;~、 *、 /、 +、 =、 ^、 |、 **、 <、 >、 <=、 =>、 == 和 <>，则必须将 *"item"* 部分括在引号中。

不过，由于某些通用 ASCII I/O 服务器接受任何字符作为有效项目名，我们极力推荐您总是将 *"item"* 部分用引号括起。

例如，我们假设您需要从 Allen-Bradley PLC 整型寄存器中获取一个比特的数据，并将 **N10:7/3** （自整数 10 起第三位）用作项目名。系统会将 **N10:** 视作 “访问名”，因为正斜杠 (/) 不是有效字符。不过，如果将项目名放在引号中，即 **"N10:7/3"**，系统会读取整个项目并将它用作项目名。

您也不能对项目名或远程标记名引用项目名进行字符串串联操作。例如，假定您使用下面的表达式创建了一个字符串输出链接：



系统执行上面的表达式时，它将使用 “访问名” **PLC2**，并通过 Allen-Bradley I/O 服务器去检索字符串文件 **ST10:1** 中存储的字符串。然后它会将字符串 **37** 追加到在 **ST10:1** 中检索的字符串的末尾。如果 "Green Paint" 存储在 **ST10:1** 中，则链接到该表达式的字符串输出对象将显示 **Green Paint37**。因此，操作员看不到他们本想要的 **ST10:137** 的内容。

每次使用远程引用 (*accessname:"item"*) 时， InTouch 会验证指定的 “访问名”。如果它确定该 “访问名” 尚未定义，则会提示您去定义它。如果在提示下选择**是**，则会出现**访问名**对话框供您添加新的 “访问名”。

远程标记名被激活时，系统也会验证 “访问名” 是否有效。如果检测到错误，它们会被写入 Wonderware Logger。

您可以删除由远程引用使用的 “访问名”，前提是本地标记名不再使用它。

如需有关定义 “访问名” 的详细信息，请参阅第 5 章 “建立分布式应用程序”。

记录远程引用标记名

缺省条件下，远程引用的标记名会记入 “历史记录” 文件。要记录远程引用标记名，您必须启用 “历史记录” 功能，然后在应用程序目录中的 INTOUCH.INI 文件中添加下面一行代码：

```
RemoteTagsLogEvents=1
```

如果不想记录 I/O 型标记名，请在应用程序目录下的 INTOUCH.INI 文件中添加下面一行代码：

```
RemoteTagsNoIOEvents=1
```

备注 仅当 **RemoteTagsLogEvents** 设置为 **1** 时，**RemoteTagsNoIOEvents** 设置才适用。

远程标记名许可证实施

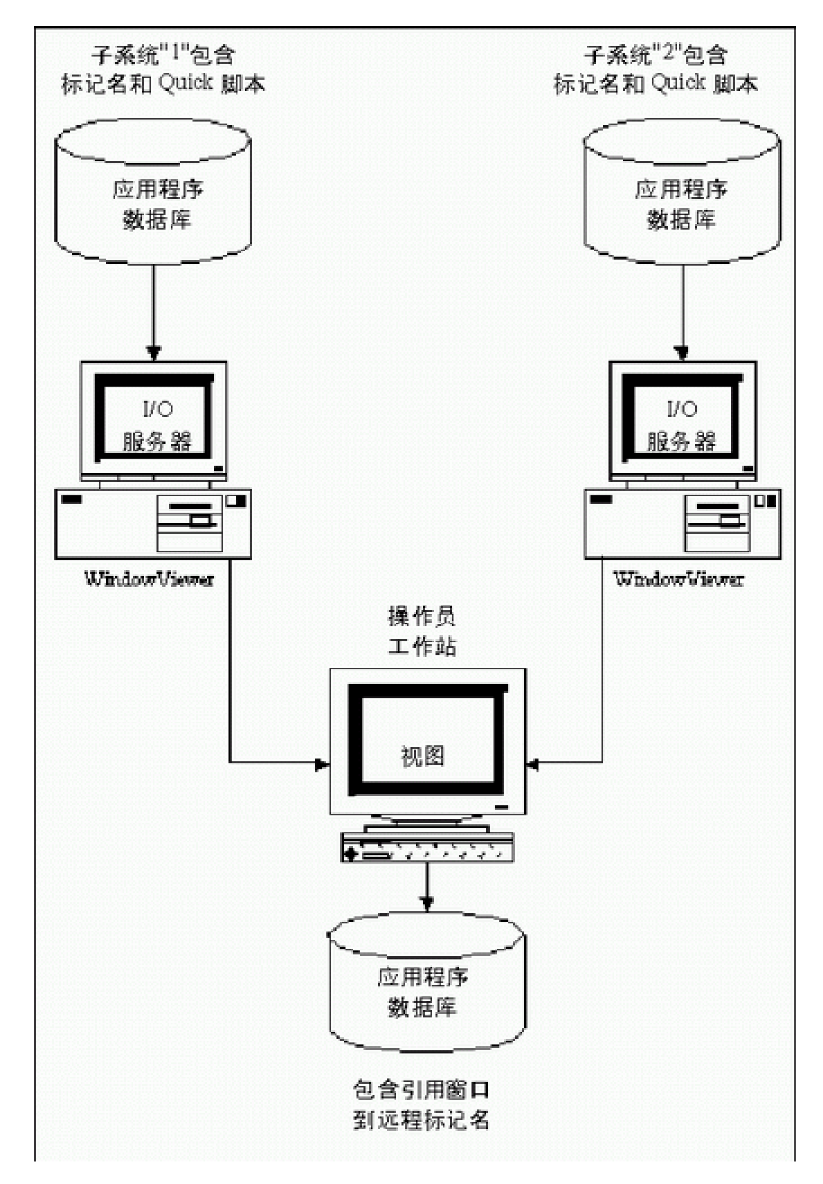
InTouch“标记名数据字典”可支持多达 61,405 个标记。 InTouch 60K 的标记名许可证可允许您使用最大数目的标记名引用。如果安装了 60K 标记名许可证，每次在窗口或 QuickScript 中激活标记名时，许可证实施逻辑所使用的标记名引用计数会递增；每次引用窗口或 QuickScript 关闭和（或）终止时，引用计数会递减。

在 InTouch 7.11 之前的版本中，在一个应用程序会话内，设计时声明的标记名总数加上有关的远程标记名引用数目必须始终小于或等于 60K。InTouch 7.11 60K 标记名许可证允许应用程序设计人员声明超过 60K 的设计时加远程标记名引用数目。远程标记名分别进入或退出应用程序会话范围时，系统会动态递增或递减总标记名引用计数。

备注 在许可证允许使用的数目小于最大标记名引用数的情况下，一旦标记名在许可限额中被引用和计数，即使不再使用该标记名，限额中的计数也不会递减。

创建标记名服务器应用程序

通过创建仅包含 InTouch QuickScript 和标记名的应用程序，您可以建立一个 WindowViewer 实例来充当标记名服务器。您可以创建另一个仅包括窗口（和用于窗口逻辑处理的内存型标记名）的应用程序。如果这些窗口仅包含远程标记名引用，则此应用程序可以充当所有过程窗口的“仓库”。在这种情况下，远程标记名引用指向充当标记名服务器的其它 WindowViewer 实例中的标记名。连接到此数据库的一个 WindowViewer 实例会充当操作员工作站。此 WindowViewer 实例可以打开任何窗口，并查看厂区任何地方的数据。例如：



远程引用对于下列对象而言是有效的：

项目	有效对象
输入链接	离散型用户输入、模拟型用户输入、字符串用户输入、垂直游标、水平游标，以及离散值按钮
离散型报警链接	线颜色、填充颜色以及文本颜色
模拟型报警链接	线颜色、填充颜色以及文本颜色
表达式	链接和脚本，其中离散、模拟或字符串型标记名可以是特定的 "向导"
数据改变脚本	"Tagname[dotfields]" （标记名 [点域]）
ActiveX 控件	事件、属性和方法
QuickScript	全部类型

远程引用对于下列对象而言是无效的：

项目	无效对象
历史趋势显示	"Pen1" 到 "Pen 8"
确认报警	（因为无法看见远程标记名进入报警状态，所以不能确认它）。

备注 对于不支持此项功能的 InTouch 先前版本创建的应用程序，不必进行转换应用程序就可以实施远程标记名引用。不过实施之后，这些应用程序将不能再与先前的版本保持向后兼容。

WindowViewer 支持 32767 个本地标记名引用和 *x* 个活动远程引用，其中 *x* = 61,405 减去在本地 "标记名字典" 中定义的标记名数。

远程引用最长可达 95 个字符。

远程引用也支持函数 **IOSetAccessName** （在 InTouch 7.0 以前版本中是 **SetDdeTopic**），此时该函数的功能与用于本地标记名时相同。

使用远程标记名引用

在客户端应用程序中，实际上您可以使用三种方法来指定远程标记名引用：

1. 在任何动画链接标记名、表达式或 InTouch QuickScript 中使用 *AccessName:"item"* 引用。

2. 导入窗口或 QuickScript，然后使用 WindowMaker **特别**菜单上的**替代标记**命令，将占位符标记名转换为远程标记名引用。

   如需有关转换占位符标记名的详细信息，请参阅 “将标记名转换为远程引用”。

   提示 InTouch 的一个强大功能是从另一个应用程序导入窗口。导入窗口时，它所有的脚本和动画链接也会一同导入。不过，动画链接和脚本中使用的所有标记名均会自动转换为占位符。您可以将所有的占位符标记名转换为远程标记名引用，并且如果需要，还可以设计一个不含本地标记名的应用程序。

   如需有关导入窗口或脚本的详细信息，请参阅第 2 章 “使用 WindowMaker”。

3. 通过在 “标记浏览器” 中将远程应用程序配置为标记源，选择要用于对象或 QuickScript 的远程标记名。例如：



如需有关在 “标记浏览器” 中选择远程标记名的详细信息，请参阅 “定义标记源”。

动态地址引用 (DRA)

“动态地址引用” 可供使用单个标记名引用多个数据源。通过将一个有效引用赋给 I/O 型标记名的 **.Reference** 点域，您就可以动态改变该标记名的数据源地址。

每个 I/O 型标记名均有一个引用与它的数据源地址关联。 **.Refrence** 点域的有效语法包括：

有效语法	描述
Tagname.Reference="accessname.item"	改变访问名和项目。
Tagname.Reference="[.]item"	相同访问名，不同项目。
Tagname.Reference="accessname."	改变访问名。
Tagname.Reference=""	使标记名无效。如果未指定 “访问名”或 “项目”，则缺省使用该点域的当前值。

备注　“动态地址引用”对于远程标记名引用而言是无效的。

如需有关 **.Reference** 点域的详细信息，请参阅联机 *Intouch 参考指南*。

使用动态引用

动态引用可用于查看那些临时需要观察值的数据点，例如在诊断应用中。因为标记名的数据源可以改变，对于那些需要永久存储或连续监视报警条件的任何数据，便不应使用动态引用。

动态引用在传统应用上一个很好的例子便是诊断应用。在此项应用中，使用单个标记名来查看 PLC 上任意模拟点的输入值。这样维护人员立即便能查看任意采样点的状态，从而实现故障诊断之目的。

要创建诊断应用程序

1. 创建一个 I/O 整型标记名。在本示例中，我们将标记名命名为 "AnalogSpy"。它的初始引用 “访问名”是 **PLC1**、项目名是 **WX001**。
2. 通过输入 # 号，创建一个文本对象。
3. 双击 # 号以打开动画链接对话框。
4. 在**用户输入**部分，单击**字符串**。此时会出现**输入 -> 字符串标记名**对话框。
5. 在**标记名**框中，输入 **AnalogSpy.Reference**
6. 单击**确定**。
7. 启动 WindowViewer 以编译和运行应用程序。

8. 单击文本对象，为赋给该标记名的 “访问名”和项目名输入一个新值。

   例如，要查看 “访问名” **PLC6** 的项目 **WX031**，请输入 **PLC6.WX031** 作为引用。

9. 如果要确认新引用是否有效，请使用下节将会介绍的 **.ReferenceComplete** 点域。

   如需有关 **.ReferenceComplete** 点域的详细信息，请参阅联机 *Intouch 参考指南*。

使用 IOSetItem 函数 改变引用

IOSetItem （在 InTouch 7.0 以前的版本中是 **SetDdeItem**）函数用于设置 I/O 型标记名的 **.Reference** 点域。此函数的基本格式是：

```
IOSetItem(TagName, AccessName, Item)
```

标记名、“访问名”和项目值可指定为文字字符串，它们也可以是其它 InTouch 标记名或函数提供的字符串值。例如，标记名 "MyTag1" 的 **.Reference** 域可通过下列语句改变为指向访问名 "Excel" 和项目 "R1C1":

```
IOSetItem("MyTag1", "Excel", "R1C1");
```

或者通过下列语句，

```
Number = 1;
TagNameString = "MyTag" + Text(Number, "#");
IOSetItem(TagNameString, "Excel", "R1C1");
```

如果给 “访问名”和项目值指定了空字符串 ("")，则该标记名将会失效。例如，使用下列语句可使标记名 "MyTag2" 失效：

```
IOSetItem("MyTag2", "", "");
```

如果只为 “访问名”值指定空字符串，则标记名的当前 “项目”值会被保留，而它的 “访问名”会被更新。例如，下列语句将标记名 "MyTag3" 的访问名改为 "Excel2"，但不影响它当前的 “项目”值：

```
IOSetItem("MyTag3", "Excel2", "");
```

与此类似，如果只给 “项目”指定了一个空字符串，则该标记名的当前 “访问名”值会被保留，而它的 “项目”值会被更新。例如，下列语句将标记名 "MyTag3" 的 “项目”改为 "R1C2"，但不影响它当前的 “访问名”值：

```
IOSetItem("MyTag4", "", "R1C2");
```

如需有关 **IOSetItem** 点域的详细信息，请参阅联机 *Intouch 参考指南*。

使用 .ReferenceComplete 确认引用

每个 I/O 型标记名均有一个 **.ReferenceComplete** 域。此离散域用于确认引用字段中所要求的项目是否反映在 **.Value** 域中。

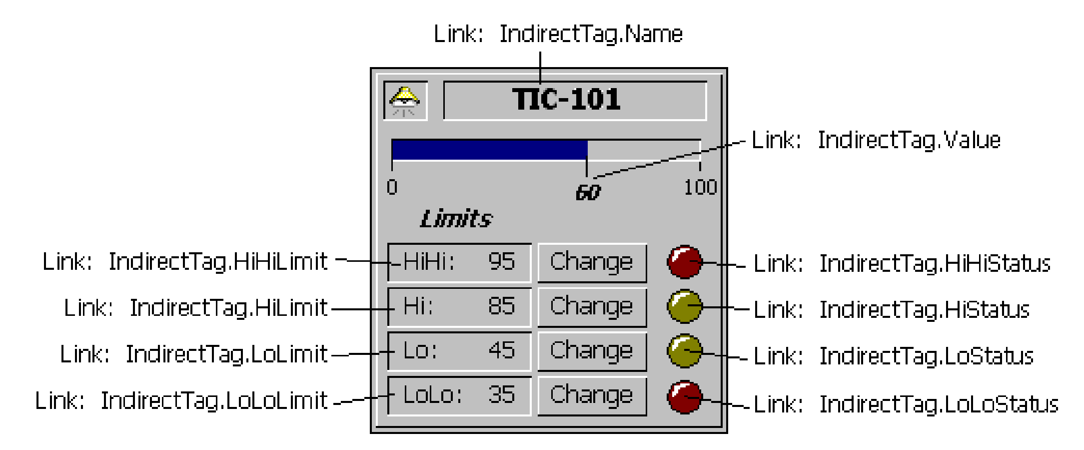
在启动 WindowViewer 时， **.ReferenceComplete** 域初始化为 false (0)。确认 **.Value** 域由 **.Reference** 域中指定的数据源更新时，**.ReferenceComplete** 值设为 true (1)。如果 **.Reference** 域发生改变，则 **.ReferenceComplete** 域会自动设置为 false (0)，然后在更新新值时，它会更新为 true (1)。

如需有关 **.ReferenceComplete** 的详细信息，请参阅联机 *Intouch 参考指南*。

在本地标记名中使用间接标记

本节介绍在本地 “标记名字典”引用中使用间接型标记名的常用方法。

间接型标记名可供创建使用多个数据集的 “通用”标记名。例如，您可以使用通用面板来修改链接至多个本地标记名的报警限。



您可以让许多不同的标记名使用您的报警限面板。要使面板转至适当的标记名，您可以执行下面的 QuickScript：

```
IndirectTagname.dotfield = "tagname";
```

其中， *tagname* 是在本地 “标记名字典”中定义的实际标记名。

执行此脚本时，与本地标记名关联的所有**点域**均可通过该间接型标记名进行访问。

如需有关标记名点域的详细信息，请参阅联机 *Intouch 参考指南*。

在远程引用中使用间接标记名

远程引用标记名在许多方式上与本地标记名存在不同。远程引用的语法是：

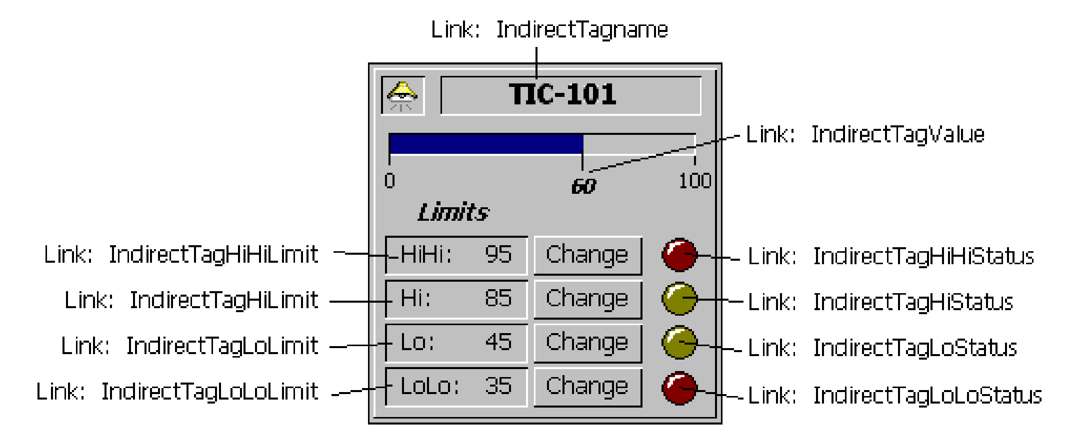
```
AccessName:Item
```

其中， ***AccessName*** 是任何有效的 InTouch “访问名”，而 ***Item*** 是 I/O 服务器 （由访问名定义）支持的任何一个有效的项目名。

如需有关远程引用的详细信息，请参阅 “远程标记名引用”。

使用远程引用时，服务器会给客户端返回一个值，而不是一个标记名结构。该值包括时间标签和质量标志。因此，指定给远程引用的间接标记名无法访问任何不与值、时间和质量有关的标记名**点域**。例如，间接标记名不能通过远程引用来访问报警限。

为解决这个问题，您可以使用多个间接标记名来创建一个面板。例如：



请注意，此面板使用 10 个间接标记名，它们都使用隐式 **.Value** 引用。

如需有关 **.Value** 的详细信息，请参阅联机 *Intouch 参考指南*。

让我们假定此报警面板被重定向至远程引用标记名 TIC-101，它位于一个名为 TagServer1 的远程 InTouch 节点上。 InTouch “访问名”已配置如下：

访问名：	TagServer1
节点名：	TagServer1
应用程序名：	View
主题名：	Tagname

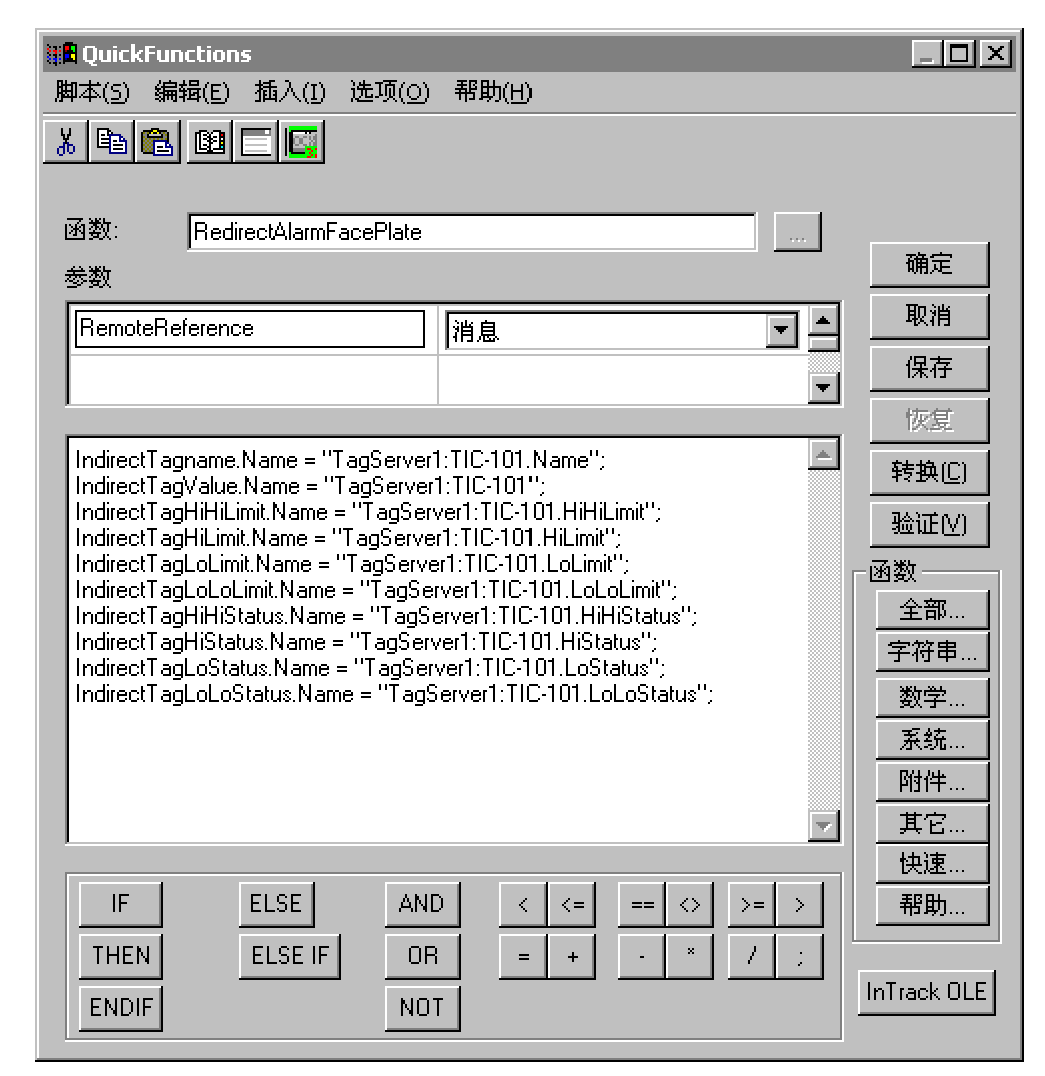
要将该面板重定向到远程引用标记名 TIC-101，请执行下面的 QuickScript：

```
IndirectTagname.Name = "TagServer1:TIC-101.Name";
```

```
IndirectTagValue.Name = "TagServer1:TIC-101";
IndirectTagHiHiLimit.Name = "TagServer1:TIC-
    101.HiHiLimit";
IndirectTagHiLimit.Name = "TagServer1:TIC-101.HiLimit";
IndirectTagLoLimit.Name = "TagServer1:TIC-101.LoLimit";
IndirectTagLoLoLimit.Name = "TagServer1:TIC-
    101.LoLoLimit";
IndirectTagHiHiStatus.Name = "TagServer1:TIC-
    101.HiHiStatus";
IndirectTagHiStatus.Name = "TagServer1:TIC-101.HiStatus";
IndirectTagLoStatus.Name = "TagServer1:TIC-101.LoStatus";
IndirectTagLoLoStatus.Name = "TagServer1:TIC-
    101.LoLoStatus";
```

由于每次面板重定向时均须执行此脚本，既浪费时间又容易出错。因此，一个更好的方法是创建 InTouch QuickFunction，可供您只写一次脚本，然后将它传递给远程引用的名称。

例如，通过使用上述的脚本命令集，您可以定义一个 QuickFunction 函数 **RedirectAlarmFacePlate()**：



现在您只需调用 QuickFunction 函数 **RedirectAlarmFacePlate()**，就可以处理整个重定向。要实现这一点，必须使用另一个 InTouch QuickScript 调用该 QuickFunction 函数。例如：

```
CALL RedirectAlarmFacePlate ("TagServer1:TIC-101");
```

通常，您应该专门使用远程引用来开发新的应用程序。（也可以远程引用 InTouch 中的本地 “标记名字典”）。这可以从图形中提取数据源（本地或远程）。通过在 InTouch 应用程序中使用专有的远程引用，以后在分布式 FactorySuite 系统中添加新的窗口或图形对象时，就可以显著减少应用程序的维护工作。

标记浏览器

“标记浏览器”是一个基本工具，可用于查看和选择本地及远程标记名和标记名**点域**，无论它们是来自 FactorySuite 应用程序，还是任何其它支持 InTouch “标记名字典”接口的标记源。标记浏览器可供选择现有的标记名、添加新的标记名，以及查看基本的 “标记名字典”信息。您也可以使用 “标记浏览器”来访问对话框，以便执行标记名的编辑和复制，以及从远程标记源中选择标记名 （远程引用）。

第一次访问 “标记浏览器”时，缺省条件下，系统会选择 **< 本地 >** 作为标记源。这表示，本地应用程序的 “标记名字典”中的标记名会显示出来。此后，屏幕将显示最近一次访问的标记源的标记名。

“标记浏览器”的操作模式有两种：“过滤选择模式”和 “无限选择模式”。“标记浏览器”的具体模式取决于您访问它的方法。下面列出了每种模式下访问 “标记浏览器”的基本方法：

无限选择模式

- 双击动画链接标记名或表达式输入框。
- 双击 ActiveX 或向导标记名或表达式输入框。
- 双击任意 InTouch QuickScript 窗口中的空白区。
- 在 InTouch QuickScript 编辑器中，选择**插入**菜单上的**标记名**命令。
- 在 InTouch QuickScript 编辑器中，按 **Alt+N** 组合键。
- 在**替换标记**对话框中，双击空白的**新名称**框。
- 在 SQL Access **绑定列表配置**中，双击**标记名点域名**输入框。

过滤选择模式

- 在 “标记名字典”中，单击**选择**按钮。
- WindowMaker 正在运行时，在 Recipe Manager **单元**定义中，双击**单元 #** 列中的单元。
- 在运行时，单击 “历史趋势设置”对话框中的任意 “笔 #”按钮。在这种情况下，“标记浏览器”将只显示用 “标记名字典”中定义时选定了 “记录数据”选项的标记名。

  提示 仅当在开发期间为历史趋势选定了**允许运行时改变**选项时，才会支持此项功能。

- 在运行时，单击链接到 **HTSelectTag()** 函数的任何对象。

如需有关 “标记浏览器”模式的详细信息，请参阅 “标记浏览器选择模式”。

“标记浏览器”的状态栏提供当前显示的标记源的下列各个项目的状态：

- 应用程序中的项目总数。
- 当前所选项目的名称。
- 所选的标记名**点域** （如果有）。
- 与标记源关联的 “访问名”。

标记浏览器选择模式

“标记浏览器”在两种选择模式下操作：“过滤选择模式”和 “无限选择模式”。

过滤选择模式

在**标记名字典**对话框中单击**选择**，或在运行时 （允许操作员在运行时改变历史趋势）为历史趋势笔选择一个新的标记名时，此模式将被激活。显示的标记名 （可供选择）仅限于当前的 InTouch 应用程序。例如：



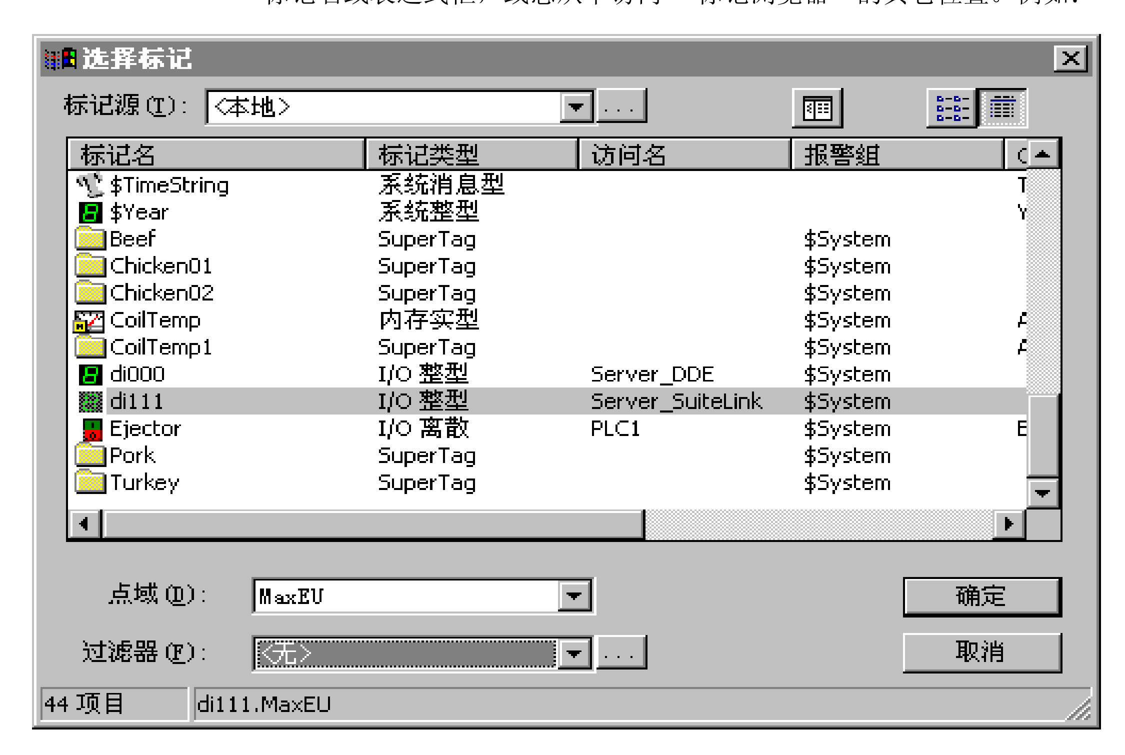
提示 从 “标记名字典”中访问 “标记浏览器”并在此视图下选择标记名时，在单击**确定**之后，会出现它的 “标记名字典”定义。

备注 标记名**点域**不能在此模式下选择。

无限选择模式

要访问无限选择模式，请双击任意 InTouch QuickScript 窗口的空白区、动画链接标记名或表达式框、**替换标记名**对话框中的空白**新名称**框。本地或远程标记源中定义的标记名可以在此模式下显示和选择。

此种模式下也可以为标记名选择标记名**点域**。在这种模式下选择标记名和（或）标记名**点域**时，它会自动输入 InTouch QuickScript、动画链接标记名或表达式框，或您从中访问 “标记浏览器” 的其它位置。例如：



要选择点域

1. 单击**点域**箭头以打开**点域**列表，从中您可以将选择点域让它与当前所选的标记名类型关联。

提示 缺省条件下，所有类型的标记名最初均显示 < 无 >。

备注 从 “标记名字典” 访问 “标记浏览器”、在运行期间从 “历史趋势设置” 对话框为历史趋势笔选择标记名时，均不能使用点域。（配置历史趋势必须使用 “允许运行时改变” 选项进行配置）。

---

2. 在列表中单击要赋给所选标记名的**点域**。

提示 并非每个标记名类型都有相同的**点域**。例如，**离散**型标记名有 **.OnMessage** 点域，而模拟型则没有。如果选择**离散**型标记名并为它指定 **.OnMessage**，然后选择另一个**离散**型标记名，则显示的**点域**列表不会改变。但是，如果在控制视图列表中选择另一种类型的标记名 （如模拟型），则显示的**点域**将恢复为 < 无 >。

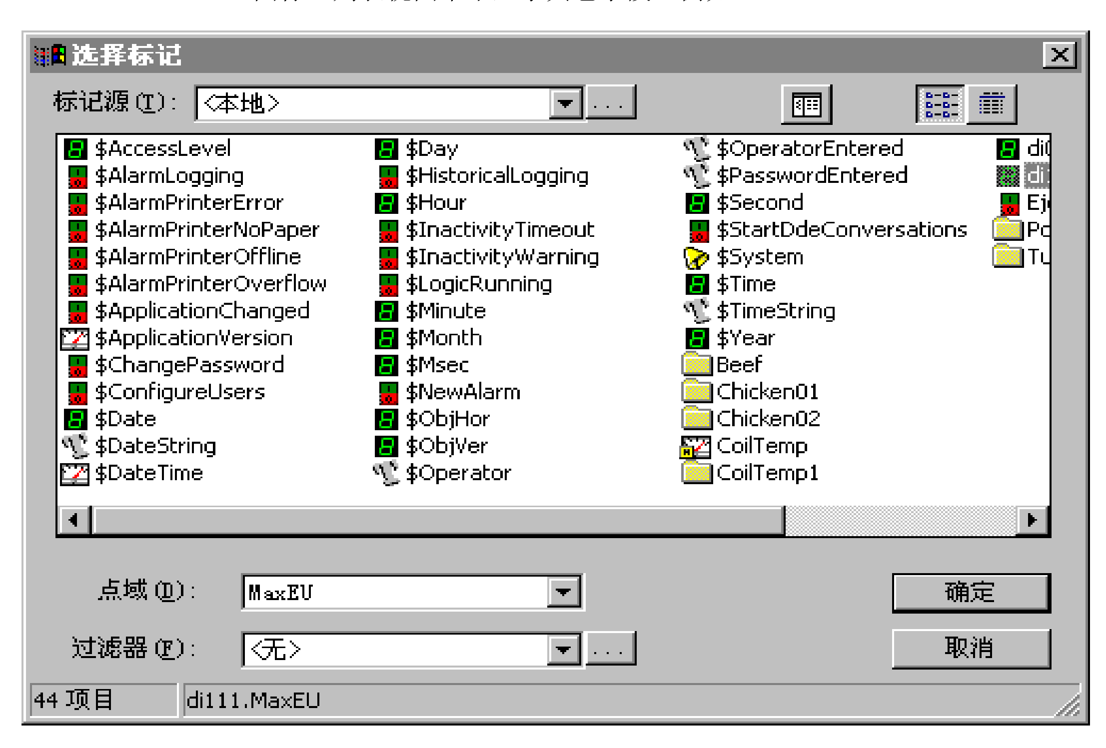
标记浏览器视图

“标记浏览器”支持三种控制视图：“标记名列表控制”、“标记名详细控制”和 “标记名树形视图控制”。

列表视图

列表视图用于在当前选择模式下显示和选择标记名 （如上所述）。“标记名列表控制”视图根据**列表视图**和**详细视图**按钮的状态，以两种视图显示标记名：

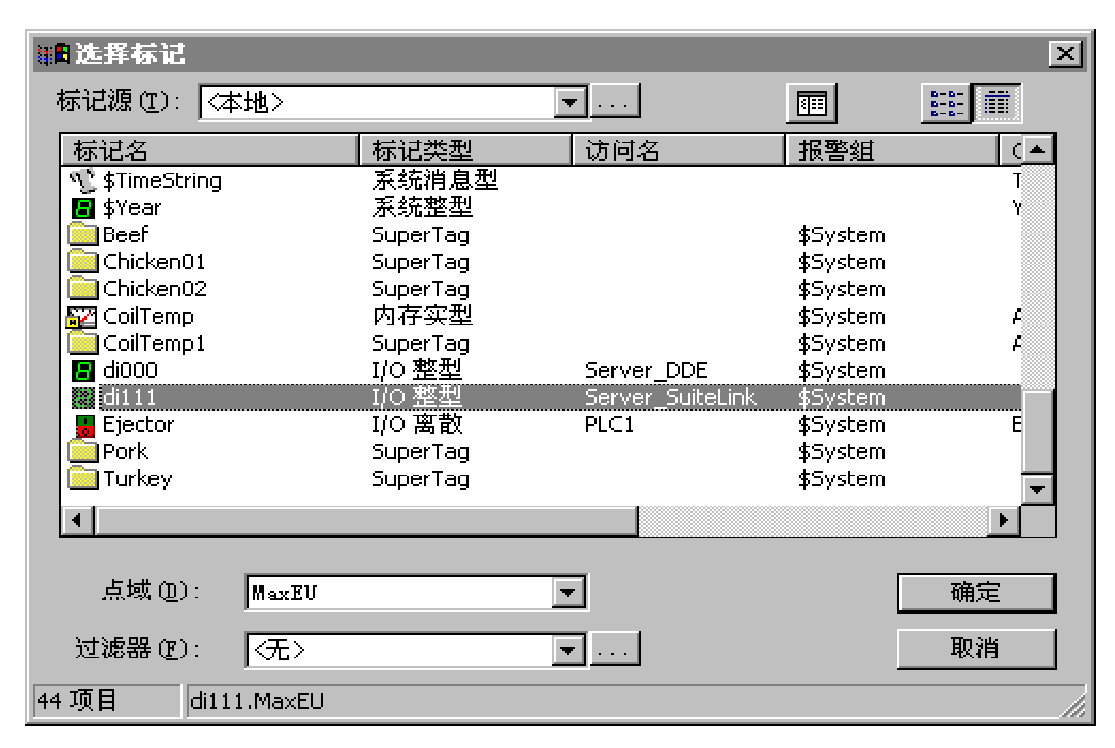
选择列表视图时，根据每个标记名的类型，标记名旁边会显示不同的小图标。列表视图中不显示其它字段。例如：



提示 要刷新 “选择标记”对话框中的显示画面，请按 F5。

详细视图

选择详细视图时，标记名及其详细资料会以多列形式进行显示。显示的详细资料包括 “标记名名称”、“标记名类型”、“访问名”、“报警组”以及 “注释”。您可以通过单击各个列的标题名，按每种详细资料类型给该列表排序。要选择某个项目，可单击该项目显示内容的一部分，而不只是标记名。（整行将辉亮显示）。例如：



提示 切换视图时，所选标记名在新视图中仍可见并辉亮显示。

树形视图

“树形视图”根据**列表视图**和**详细视图**按钮的状态，以两种视图显示标记名。选择树形视图时，对话框的左侧会出现一个窗格。通过使用 “树形视图”，您也可以访问任何 SuperTag 模板中的成员标记名。

选择“树形视图”时，如果**详细视图**模式处于活动状态，则“标记浏览器”显示如下：

提示 要在“树形视图”下展开列表，请双击应用程序名，或单击 ⊞。要折叠列表，请再次双击应用程序名，或单击 ⊟。在树形视图窗格中双击应用程序与在**标记源**列表中选择它效果相同。

备注 在“标记浏览器”中向下展开不同的层次时，您可以使用 BACKSPACE 键“返回”上一层。

定义标记源

您必须在“标记浏览器”中定义要查看的标记源。本节介绍有关添加、删除或编辑标记源的操作程序。添加或编辑标记源定义时，您可以输入各种信息，如要与标记源的标记名关联的本地网络“访问名”、用户定义的应用程序名、标记源的数据源等。

备注 在将占位符标记名转换为远程标记名引用时，您也会用到这些操作程序。

如需有关远程标记名引用的详细信息，请参阅“将标记名转换为远程引用”。

要定义标记源

1. 打开“标记浏览器”，然后单击“定义标记源”按钮。此时会出现**定义标记源**对话框。



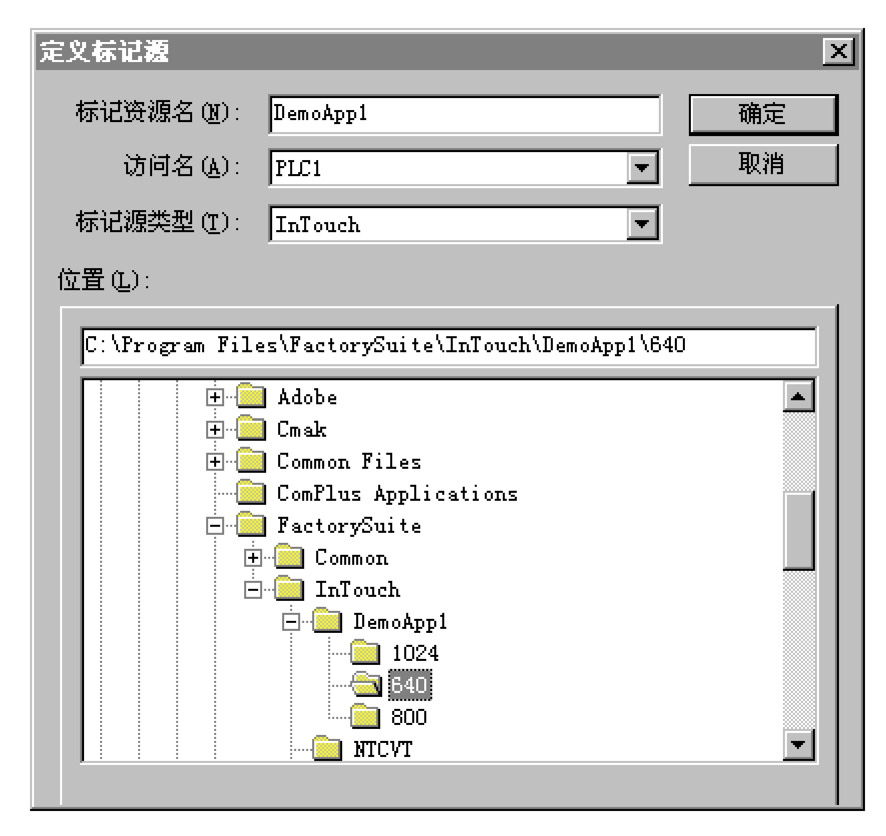
备注 如果标记源已经定义，则对话框出现时，它们会显示在其中。列表将包含标记源的用户定义名称、标记源的位置（路径）以及与应用程序关联的本地网络“访问名”。

提示 要选择多个标记源，请在单击每个名称的同时按下 SHIFT 键。要选择列表中不连续的多个标记源，请在单击每个名称的同时按下 CTRL 键。

备注 **定义标记源**对话框关闭时，您必须单击“标记浏览器”中的**标记源**箭头，并从列表中选择新的标记源。随后“标记浏览器”会刷新并显示所选标记源的标记名。

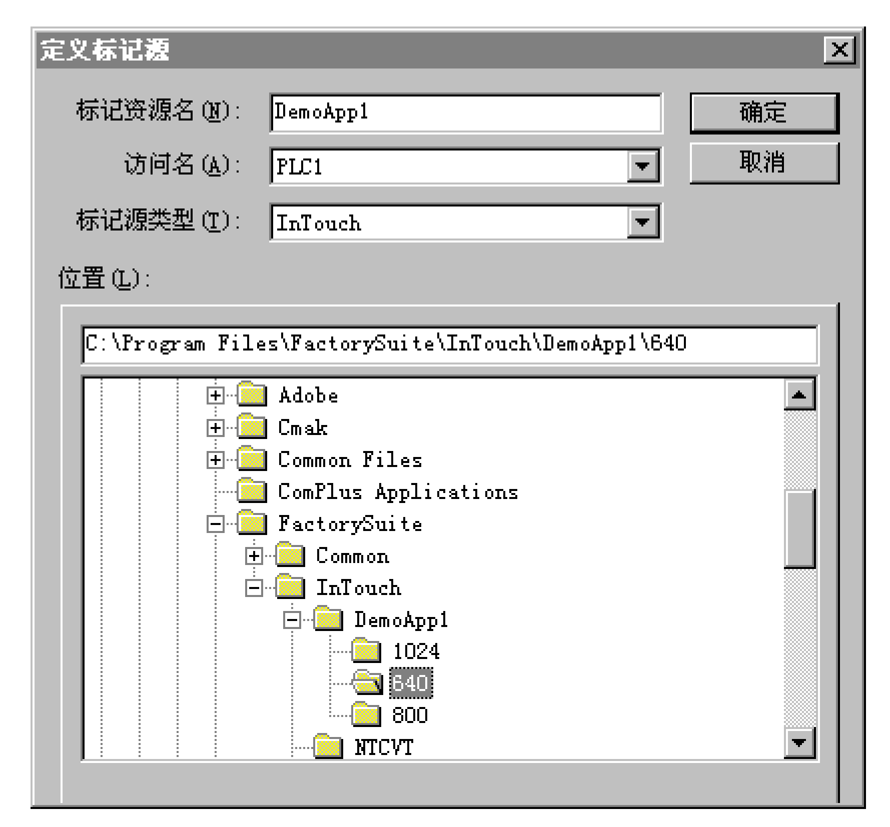
2. 要从“标记浏览器”的**标记源**列表中移除某个标记源，请单击“定义标记源”按钮。此时会出现**定义标记源**对话框。从列表中选择标记源，然后单击**删除**。
3. 要编辑已定义的标记源，请从列表中选择它，然后单击**编辑**。此时会出现**定义标记源**对话框，显示所选标记源的配置。
4. 要定义新的标记源，请单击**新建**。此时会出现**定义标记源**对话框。

备注 单击“新建”时，如果本地应用程序中没有定义“访问名”，则会出现一个消息框，告诉您没有定义“访问名”，因此不允许定义新的标记源。（标记源必须与本地网络“访问名”关联）。



5. 在**标记源名**框中，输入用于标识标记源的名称。
6. 单击**访问名**箭头，选择本地应用程序中要与标记源中的标记名关联的“访问名”。
7. 单击**标记源类型**箭头，并为该标记源的标记名数据库选择来源。（缺省条件下，显示的是 **InTouch**）。
8. **位置**框会显示标记源的完整路径。

9. 在目录树窗格中，找到标记源，然后单击**确定**。此时会再次出现**定义标记源**对话框，显示所选的标记源：



10. 单击**关闭**。此时 “标记浏览器”会再次出现。

11. 单击 工具以显示树形视图窗格，让已定义的所有标记源均显示出来：

提示 如果未在使用树形视图模式，请单击**标记源**箭头，并选择要显示在列表中的标记源的名称。此时 “标记浏览器”会刷新并显示标记源的标记名。

备注 第一次访问 “标记浏览器”时，缺省情况下，系统会选择 < **本地** > 作为**标记源**。此后，屏幕上将显示上次访问的标记源的标记名。

12. 单击**确定**。

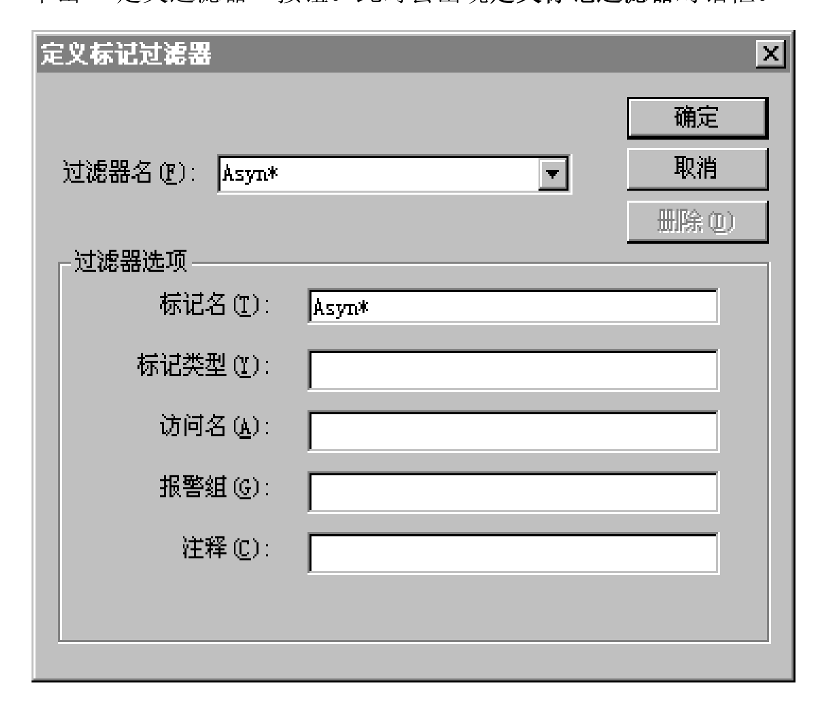
定义标记浏览器过滤器

您将使用本节介绍的操作程序去定义要用于填充 “标记浏览器”的过滤器 （搜索标准）。通过创建过滤器，您可以对任何标记名列表进行排序，并只显示符合指定标准的标记名。您可以根据**标记名**、**标记名类型**、**访问名**、**报警组**以及标记名**注释**对标记名进行排序。您可以使用这些项目中的一个或它们的任意组合来设置显示标准。您也可以保存每个过滤器实例，以便随时重新使用。

提示 例如，如果 “标记名字典”中定义了 40,000 个标记名，但只需处理其中 20 个左右赋给特定 “访问名”或 “报警组”的标记名，则此时您就可以创建一个过滤器，指定 “访问名”和 （或）“报警组”作为搜索标准，标记名必须满足此标准才会显示在 “标记浏览器”中。

要定义搜索过滤器

1. 单击 “定义过滤器”按钮。此时会出现**定义标记过滤器**对话框。



> **提示** 如果右击任意文本输入框，则会出现一个菜单，显示出您可以给所选文本应用的各个命令。

2. 在**过滤器名**框中，为要定义的过滤器输入一个唯一的名称，或单击**过滤器名**箭头，以便从列表中选择一个先前定义的过滤器名称。（在定义过滤器时，输入的**过滤器名**会添加到列表中）。

> **提示** 所有**过滤器选项**控件 （**标记名**、**标记类型**、**访问名**、**报警组**以及**注释**）均允许您输入一个通配符表达式，以限制搜索范围。如果不使用过滤器，则屏幕会显示当前所显示标记源的所有标记名。
>
> 多字符通配符是星号 (*)。例如， "Asyn*" 将搜索所有以 "Asyn" 开头的标记名。
>
> 单字符通配符是问号 (?)。例如，过滤器 "Tag?" 将搜索所有以 "Tag" 开头的四个字符的标记名。过滤器 "Tag??" 将搜索所有以 "Tag??" 开头的五个字符的标记名，等等。
>
> 过滤器可以使用有效的 InTouch 标记名字符与两个通配符的任意组合。有效的标记名字符有：A-Z、 a-z、 0-9、 !、 @、 -、 ?、 #、 $、 %、 _、 \ 以及 &。

3. 在**标记名**框中，输入要用作过滤器的标记名表达式。如果保留为空白，则系统将在过滤器定义中忽略此字段。
4. 在**访问名**框中，输入要用作过滤器的本地 “访问名”表达式。如果保留为空白，则系统将在过滤器定义中忽略此字段。

5. 在**报警组**框中，输入要用作过滤器的 “报警组”表达式的名称。如果保留为空白，则系统将在过滤器定义中忽略此字段。
6. 在**注释**框中，输入要用作过滤器的注释表达式。如果保留为空白，则系统将在过滤器定义中忽略此字段。
7. 单击**确定**以关闭对话框。

提示 现在**过滤器名**将出现在 “标记浏览器”的**过滤器**列表中，您可以选择它以便只显示与过滤器中指定的标准相符的标记名。

要删除搜索过滤器

1. 单击**过滤器**箭头，从列表中选择要删除的过滤器名。
2. 单击**删除**。此时该过滤器会被立即删除。

InTouch 交叉引用实用程序

“标记名交叉引用”实用程序可供您确定动画链接、向导、 InTouch QuickScript、 QuickFunction、 ActiveX 控件、脚本以及下列 InTouch 附加程序中标记名、远程标记名以及 SuperTag 的使用情况：SPC Pro、 SQL Access Manager 以及 Recipe Manager。对于所有的对象 （如向导、ActiveX 控件和动画链接），它显示窗口名和链接到该标记名的所有对象的坐标。它还可供查看标记名所在的任何 QuickScript 或 QuickFunction。

提示 为方便起见，在您执行其它任务时，“标记名交叉引用”实用程序可以在 WindowMaker 中保持打开。

要使用 **InTouch** 交叉引用实用程序

1. 在**特别**菜单上，单击**交叉引用**，或者在 “应用程序浏览器”中，双击**交叉引用**。此时会出现 **Intouch 交叉引用搜索标准**对话框。



2. **搜索标准**组可供限制搜索范围。通过只选择所需的选项，您可以轻松确定该范围。

搜索所有实例	搜索标记名或 SuperTag 在动画链接、InTouch QuickScript 以及所有的附加程序 （如 SPC、 SQL Access Manager、Recipe Manager）中的所有使用情况。
搜索特定实例	仅搜索满足指定选项的标记名或 SuperTag。例如，如果您只想搜索在窗口脚本中的使用情况，则可以只选择在窗口脚本中的使用情况。

3. 在**过滤器**框中，输入标识所定义的过滤器的唯一名称，或单击**过滤器**箭头，以便从列表中选择先前定义的过滤器。（在定义过滤器时，您输入的名称将添加到**过滤器**列表中）。

提示 过滤器编辑器控件可供输入通配符表达式来限制标记名的搜索范围。如果不使用过滤器，则会检索当前应用程序中所有标记名的信息。

多字符通配符是星号 (*)。例如， "Asyn*" 将搜索所有以 "Asyn" 开头的标记名。

单字符通配符是问号 (?)。例如，过滤器 "Tag?" 将搜索所有以 "Tag" 开头的四个字符的标记名。过滤器 "Tag??" 将搜索所有以 "Tag??" 开头的五个字符的标记名，以此类推。

过滤器可以使用有效的 InTouch 标记名字符与两个通配符的任意组合。有效的标记名字符有：A-Z、 a-z、 0-9、 !、 @、 -、 ?、 #、 $、 %、 _、 \ 以及 &。

如果在**过滤器**框中右击鼠标，则会出现一个菜单，显示可应用于所选文本的各条命令。

4. 单击**搜索**，根据指定的查看标准开始交叉引用搜索。

查看交叉引用搜索结果

执行交叉引用搜索时，会出现 **InTouch 交叉引用实用程序**对话框，列出根据您指定的**过滤器**找到的所有使用实例。

如果没有使用过滤器，则会显示当前应用程序的 “标记名字典”中定义的所有标记名。例如：



交叉引用实用程序图标

下面简要介绍可能出现在 “InTouch 交叉引用实用程序”中的各种图标：

图标名称	图标	描述
展开层视图	⊞	标记名或 SuperTag 被赋给 InTouch 对象，或用于存储 InTouch QuickScript、向导或附加程序中的值。单击此图标可以展开层视图。
折叠或展开层视图	⊟	单击此图标可以折叠被展开的层视图。
未指定给对象		显示的标记名或 SuperTag 是在应用程序的 “标记名字典”中定义的，但它不能赋给对象。
双击以展开视图		显示的标记名或 SuperTag 可用在动画链接或 InTouch QuickScript 中。双击或单击 ⊞ 可以展开视图。
双击以显示窗口名		显示的标记名或 SuperTag 被赋给动画链接。双击或单击 ⊞ 可以显示窗口名和赋给动画链接的对象在窗口中的坐标。
双击以展开视图及显示类型		显示的标记名或 SuperTag 用在应用程序脚本中。双击或单击 ⊞ 可以展开视图，并显示使用该标记名或 SuperTag 的应用程序脚本的类型。
全部显示		显示所有应用程序**启动时**、**运行期间**和**关闭时**脚本；窗口**显示时**、**显示期间**和**隐藏时**脚本；按键的**按下时**、**按下期间**和**放开时**脚本。双击脚本可以查看它。
双击以显示窗口名及脚本		显示的标记名或 SuperTag 用在窗口脚本中。双击或单击 ⊞ 可展开视图以显示脚本和窗口名。双击任意列出的窗口名可以查看脚本。
双击以展开数据改变脚本视图		显示的标记名或 SuperTag 用在数据改变脚本中。双击或单击 ⊞ 可以展开视图，然后双击任何列出的脚本便可查看它。
双击以展开视图，显示脚本的条件 / 类型		显示的标记名或 SuperTag 用在条件脚本中。双击或单击 ⊞ 可以展开视图并显示该脚本的条件和类型。例如， **$Hour==12 On True**。双击任何所列脚本便可查看它。

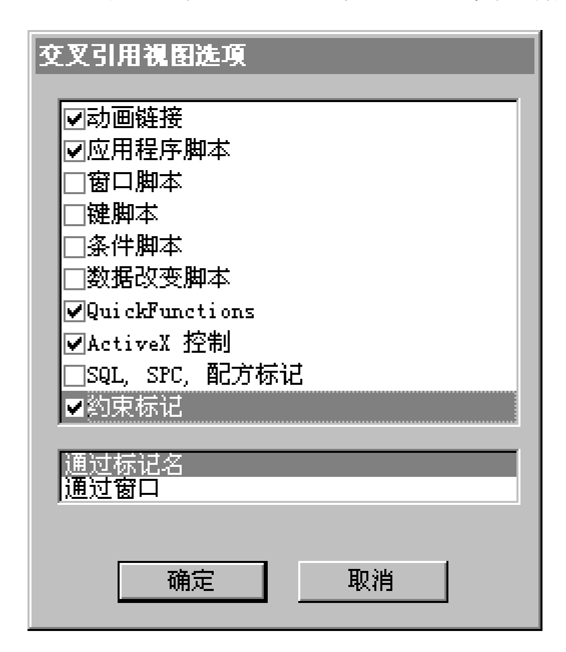
图标名称	图标	描述
双击以展开视图并显示分配给脚本 / 脚本类型的键		显示的标记名或 SuperTag 用在键脚本中。双击或单击 ⊞ 可以展开视图并显示赋给脚本和脚本类型的键。例如， **F2 On Key Down**。双击任何所列脚本便可查看它。
双击以展开视图并显示 QuickFunction		显示的标记名或 SuperTag 用在 QuickFunction 中。双击或单击 ⊞ 可以展开试图并显示使用该标记名或 SuperTag 的 QuickFunction。单击以展开视图并显示使用该标记名或 SuperTag 的 QuickFunction 的名称。双击任何所列脚本便可查看它。
双击以展开试图并显示 ActiveX 事件脚本		显示的标记名或 SuperTag 用在 ActiveX 事件脚本中。双击或单击 ⊞ 可以展开视图并显示 ActiveX 事件脚本。
按窗口交叉引用时使用		按窗口交叉引用时，此图标显示在使用了所显示的标记名或 SuperTag 的窗口名之前。双击或单击 ⊞ 可以查看窗口中使用的所有标记名。
显示的标记名或 SuperTag 用在 SPC Pro 应用程序中		显示的标记名或 SuperTag 用在 SPC Pro 应用程序中。双击或单击 ⊞ 可以查看其中使用了标记名或 SuperTag 的 “SPC 数据集” 的名称。
显示的标记名或 SuperTag 用在 SQL 应用程序中		显示的标记名或 SuperTag 用在 SQL 应用程序中。双击或单击 ⊞ 可以查看其中使用了标记名或 SuperTag 的 “SQL 绑定列表”。
显示的标记名或 SuperTag 用在 Recipe Manager 应用程序中		显示的标记名或 SuperTag 用在 Recipe Manager 应用程序中。
显示的标记用作报警约束。		显示的标记用作报警约束。双击或单击 ⊞ 可以查看作为约束的所有标记的名称。

改变交叉引用搜索标准

如有必要，在完成最初的交叉引用搜索之后，您可以通过修改原始搜索选项来缩小搜索范围。

要改变搜索选项

1. 在 **InTouch 交叉引用实用程序**对话框中 （执行初始搜索后显示的），单击**选项**。此时会出现 **InTouch 交叉引用视图选项**对话框。



2. 选择要修改以用于新搜索的搜索标准选项。

提示 此处可用的选项取决于您原先在 **InTouch 交叉引用搜索标准**对话框中所选的**搜索标准**。如果您选择了**搜索所有实例**，则所有的搜索标准选项都可用。如果您选择了**搜索特定实例**，则只有您指定的那些选项可用。要改变**搜索标准**选择，请单击**取消**。此时会再次出现 **InTouch 交叉引用实用程序**对话框。单击**搜索**，并选择新的**搜索标准**选项。

3. 在对话框底部的列表中，选择要在树形视图中显示标记名还是窗口名，然后单击**确定**。

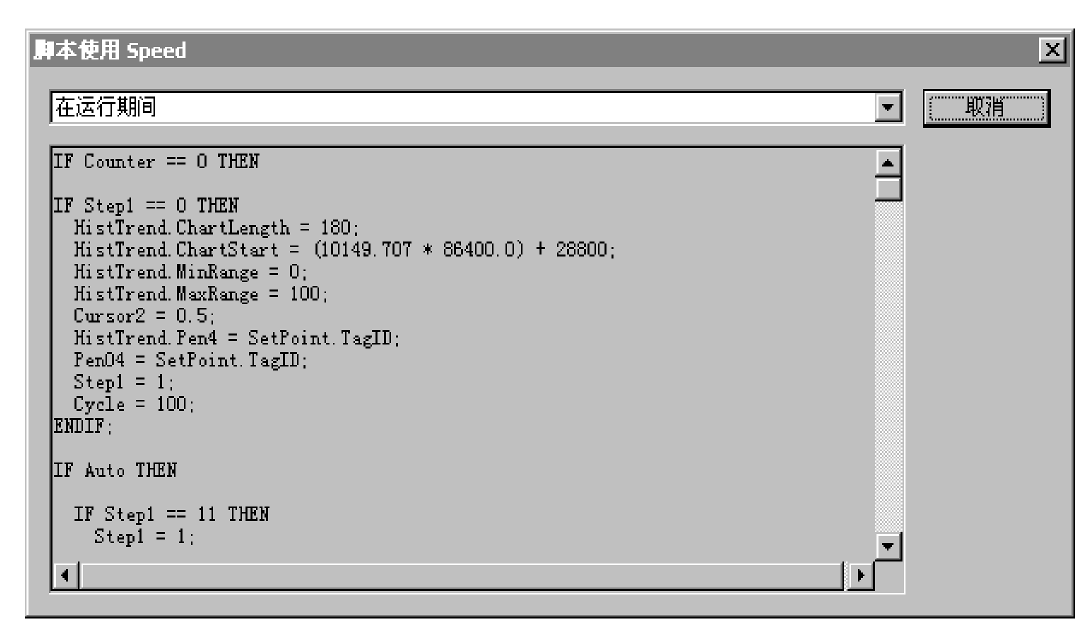
按标记名交叉引用

按字母顺序列出根据指定的搜索标准找到的所有标记名 （缺省视图）。根据指定的搜索标准，此视图可供查看在窗口、动画链接、脚本和附加应用程序中找到的所有标记名的使用情况。

提示 您可以双击所显示的标记名，然后双击**动画链接使用**来展开视图。展开该视图时，会显示窗口名和链接到该标记名的对象位置（坐标）。例如：



此外，您还可以双击标记名，然后再双击与之关联的任何脚本，以便在**脚本使用***标记名*对话框中打开它：



屏幕顶部的列表框显示了与所选标记名关联的所有脚本。单击箭头打开列表以选择要查看的其它脚本。对于 “应用程序”、“窗口”、“键”和“条件脚本”，该列表将包含所有使用此标记名的脚本的名称。对于“数据改变”脚本，则只会列出标记名。对于 QuickFunction，该列表将包含所有 QuickFunction 的名称。

备注 这是 QuickScript 的只读视图。您不能在此对话框中编辑 QuickScript 文本。不过，您可以复制 QuickScript 任一部分或全部，然后将它粘贴到任何 InTouch QuickScript 编辑器窗口中。

要将 QuickScript 复制到 Windows “剪贴板”，请右击该脚本，然后单击**全部选择**。再次右击该脚本并单击**复制**。您也可以执行 Windows 复制命令 (**Ctrl+C**)。

要将复制的脚本粘贴到另一个 InTouch QuickScript 中，在**脚本**下的 “应用程序浏览器”中，双击要创建的脚本类型。此时会出现 QuickScript 编辑器。在**编辑**菜单上，单击**粘贴**，或右击脚本窗口，然后单击**粘贴**。您也可以运行 Windows 的粘贴命令 (**Ctrl+V**)。

单击**取消**以关闭该对话框。

按窗口名交叉引用

先按窗口名，然后按窗口中的标记名给显示对象进行排序。例如：



备注 此视图只显示窗口中使用的标记名，它不包含在动画链接、脚本等环境中的使用情况。

单击**展开视图**，显示当前所显示的标记名或窗口的所有可用视图级别。例如：



单击**收敛视图**让对话框返回缺省模式。

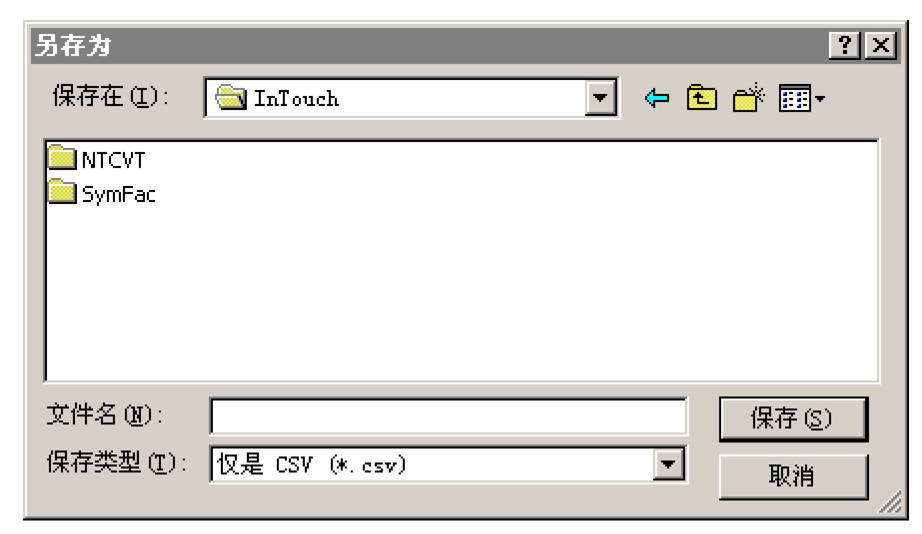
单击**关闭**以退出交叉引用实用程序。

保存交叉引用文件

交叉引用文件可以保存，并稍后在支持逗号分隔变量 (.CSV) 文件的任意文本编辑器程序中进行查看。存储在交叉引用文件中的信息对应于当前显示在 **InTouch 交叉引用实用程序**对话框中的信息。

要保存交叉引用文件

1. 在 **InTouch 交叉引用实用程序**对话框中，单击**另存为**。此时会出现**另存为**对话框。



2. 在**文件名**框中，输入要用以保存交叉引用文件的名称。

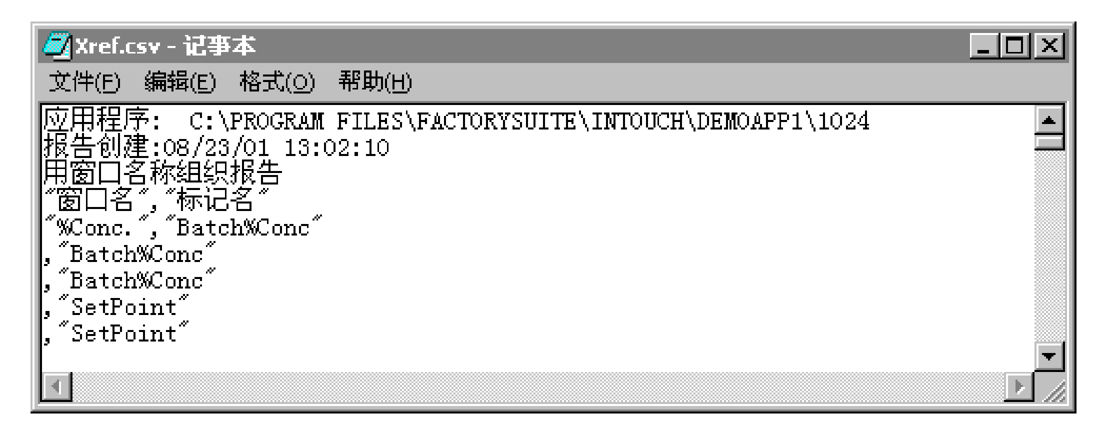
备注 该文件<u>必须</u>以 .csv 格式保存。

3. 单击**保存**。

打印交叉引用文件

在任何一个支持 .CSV 文件格式的文本编辑器程序中，您可以打开交叉引用 .CSV 文件，并将它作为报告进行打印。

例如，如果您在 “记事本” 中打开文件，则它会显示如下：



要在 “记事本” 中打印该文件，请在**文件**菜单上单击**打印**。

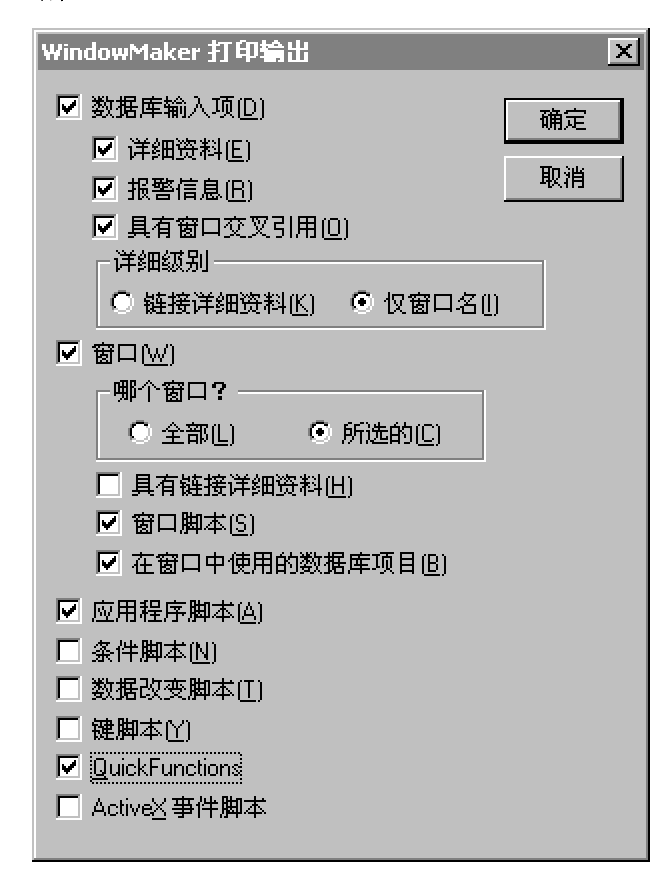
打印标记名字典详细资料

除了打印已保存的交叉引用 .csv 文件外，您还可以打印“标记名字典”详细资料、报警信息、链接详细资料以及脚本的列表。打印“标记名字典”详细资料有助于您确定标记名的使用状况。

备注 Windows 缺省打印机将用于产生宽 80 列的打印输出。缺省打印机的选择和设置均通过 Windows“控制面板”来进行。

要打印标记名字典详细资料

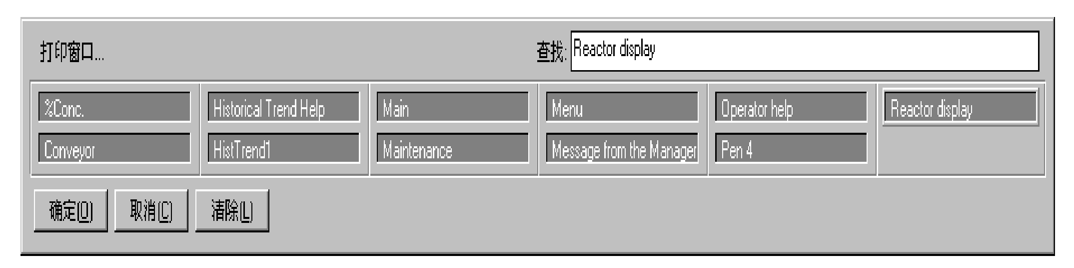
1. 在**文件**菜单上，单击**打印**。此时会出现 **WindowMaker 打印输出**对话框。



2. 如果要打印所有的数据库信息，请选择**数据库输入项**。如果选择**数据库输入项**，则下面三个选项会被激活：
   - 选择**详细资料**以便在报告中包括数据库详细资料。
   - 选择**报警信息**以便在报告中包括数据库报警信息。
   - 选择**具有窗口交叉引用**以打印所有包含窗口交叉引用的数据库项目。选择此选项将激活**详细级别**选项：

- 选择**链接详细资料**以打印使用标记名的位置和动画链接详细资料。
- 选择**仅窗口名**以便只打印交叉引用窗口的名称。

3. 选择**窗口**以打印在应用程序窗口中使用的数据库输入项列表。如果您选择**窗口**，则下面三个选项会被激活：
   - 选择**全部**以打印应用程序中所有窗口的数据库输入项。
   - 选择**所选的**以只打印指定窗口的数据库输入项。此时会出现**打印窗口**对话框。



4. 选择要打印的窗口，然后单击**确定**。（缺省条件下，对话框出现时，所有的窗口名均被选定）。
   - 选择**具有链接详细资料**以打印窗口的链接详细资料。
   - 选择**窗口脚本**以打印与窗口关联的脚本。
   - 选择**在窗口中**使用的**数据库项目**以打印窗口中使用的标记名。
   - 选择**应用程序脚本**以打印应用程序脚本。
   - 选择**条件脚本**以打印与窗口关联的脚本。
   - 选择**数据改变脚本**以打印与窗口关联的数据改变脚本。
   - 选择**键脚本**以打印与窗口关联的脚本。
   - 选择 **QuickFunction** 以打印 QuickFunction。
5. 单击**确定**以开始打印报告。

从字典中删除标记名

InTouch 为数据库中的每个项目维护一份使用计数。对于特定的一些操作，如删除窗口、改变链接或脚本中的标记名等，此计数不会自动更新。在这些情况下， InTouch 认为该标记名继续由应用程序使用，因此不允许删除它。所以，要删除该标记名，您需要更新使用计数。

要删除未使用的标记名

1. 如果 WindowViewer 正在运行，请关闭它。
2. 在**特别**菜单上，单击**标记名字典**。此时会出现**标记名字典**对话框。
3. 单击**选择**。此时会出现**选择标记名** （标记浏览器）。
4. 选择要删除的标记名，然后单击**确定**。此时会出现**标记名字典**对话框，显示所选标记名的定义。
5. 单击**删除**。

备注 如果 WindowViewer 正在运行或 InTouch 认为该标记名正在应用程序中使用，则**删除**按钮不可用。

通过 InTouch 交叉引用实用程序 （在**特别**菜单上，单击**交叉引用**），您可以确定标记名的使用位置；或者您可以打印窗口中使用的所有标记名链接报告。（在**文件**菜单上，单击**打印**）。

若您确定要删除的标记名未被使用，并且**删除**按钮不可用，可打开 WindowMaker，当显示时在**特别**菜单上单击**更新使用计数**及**是**，然后单击**确定**。

如需有关打印报告的详细信息，请参阅 “打印标记名字典详细资料”。

更新使用计数

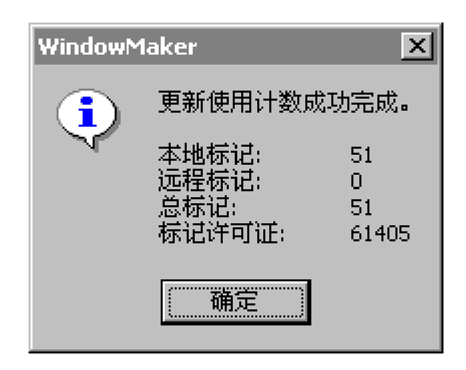
因为 InTouch 会为数据库中的每个项目维护一份使用计数，所以您需要更新使用计数，将不再使用的所有标记名设置为零，之后 InTouch 才允许您删除它们。

要更新标记名使用计数

1. 关闭所有的窗口。
2. 在**特别**菜单上，单击**更新使用计数**。

提示 有一个消息框会显示出来，告诉您更新使用计数可能要持续一段时间。此时，您可以取消命令或继续执行。

3. 单击**是**继续更新使用计数。在系统更新使用计数完毕之后，会出现下列对话框。



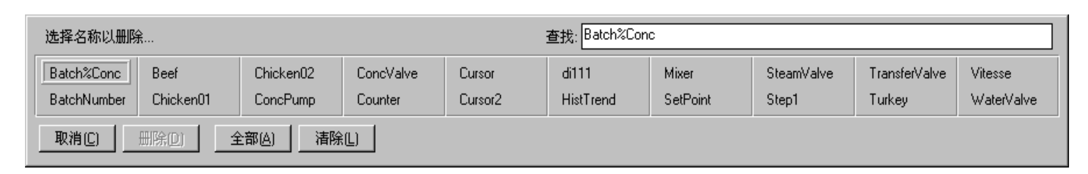
4. 单击**确定**。

删除未使用的标记名

在完成更新使用计数后， InTouch 将允许删除所有未使用的标记名。您可以通过在**标记名字典**对话框中打开每个标记名，然后单击**删除**来将其删除，也可以通过使用**删除未使用标记**命令一次删除一个或多个标记名。

删除多个未使用的标记名

1. 在**特别**菜单上，单击**删除未使用标记**。此时会出现**选择名称以删除**对话框。



2. 选择要删除的标记名，然后单击**删除**。
3. 单击**全部**以删除显示的所有标记名。

> **注意！**报警状态的标记名没有使用计数，因此可能会被意外删除。要确保将报警状态的标记名包括在使用计数中，您需要在窗口或 QuickScript 中使用它们。

显示标记名使用计数

您可以在 WindowMaker 菜单栏中显示 “标记名字典”中当前定义的本地标记名数。（标记名计数不包括内部系统标记名或远程标记名引用）。

要显示标记名计数

1. 在**特别**菜单上，指向**配置**，然后单击 **WindowMaker**。此时会出现 **WindowMaker 属性 - 通用**属性页。

提示 要快速访问该对话框，请在**配置**下的“应用程序浏览器”中，双击 **WindowMaker**。



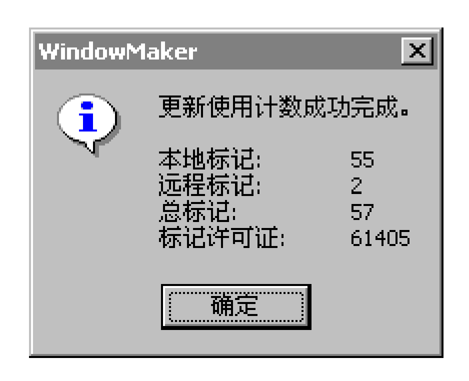
2. 选择**显示标记计数**。
3. 单击**确定**。
4. “标记名字典”中定义的本地标记名数会显示在 WindowMaker 菜单栏的一端。

提示 要更新所显示的标记名计数，必须读取整个“标记名字典”。因此，在启用此选项时，如果对“标记名字典”进行更改，则性能可能会下降。如果“标记名字典”较大，则不应选择此选项。

要确定远程标记名的使用情况

1. 在“特别”菜单上，单击“更新使用计数”。
2. 系统将更新标记名的使用情况，然后显示下列对话框：

提示 更新使用计数可能需要持续一段时间。



3. **远程标记**行将显示应用程序中使用的远程标记名数。
4. 单击**确定**。

替换标记名

复制对象时，所得复件将是原件的完全拷贝，包括链接、动画、脚本等等。不过，如果要在所复制的对象上使用不同的标记名，则必须改变该标记名。在 WindowMaker 中，这就是所谓的 “替换标记名”。您可以随时选择和改变任何对象的标记名，也可以选择多个对象并同时更改它们的标记名。

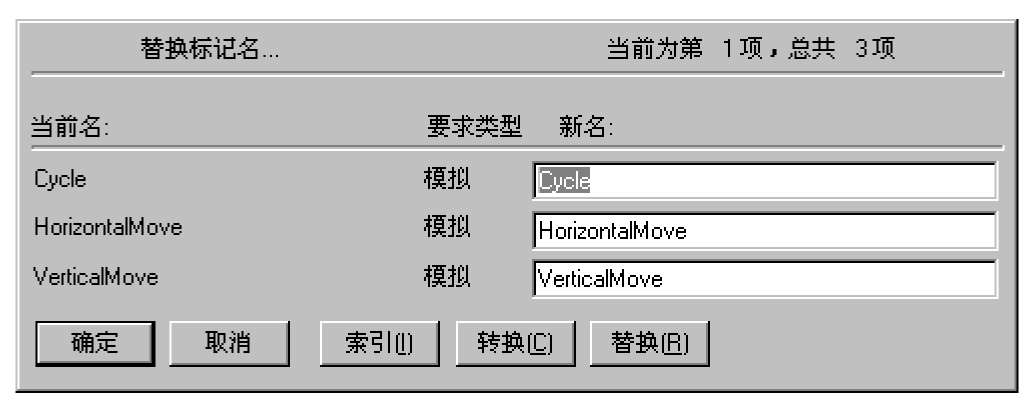
提示 如果在 WindowViewer 运行时改变对象的标记名，则只有在重新启动 WindowViewer 之后，所作改变才会生效。

如果系统许可证支持有限数目的标记名，则您也可以将本地标记名转换为远程引用标记名，以减少本地 “标记名字典”中定义的标记名数。

要将对象的标记名改变为另一个本地标记名

1. 选择要改变其标记名的对象，然后在**特别**菜单上，单击**替代标记**。此时会出现**替换标记名**对话框。

提示 要快速访问该对话框，请右击所选对象之一，指向**替换**，然后单击**替代标记**。



2. 在**新名称**对话框中，输入新的标记名，然后单击**确定**。与所选对象关联的标记名将会自动改变。

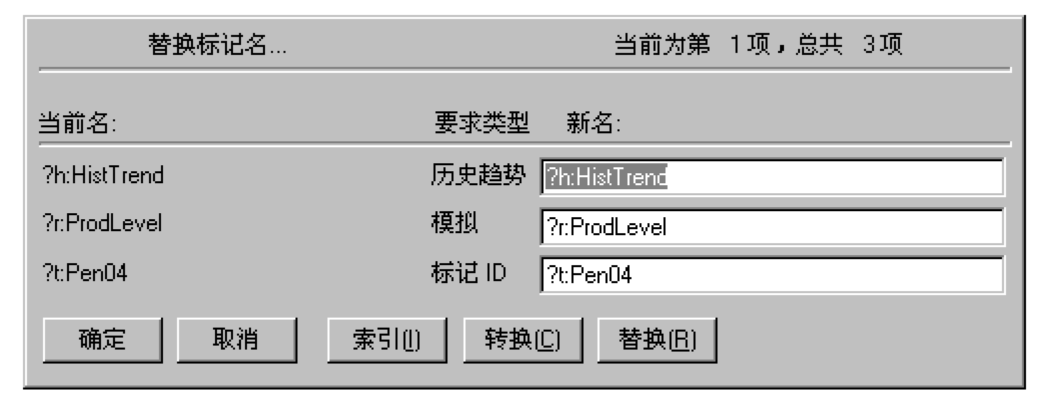
提示 如果右击**新名称**框，会会出现一个菜单，显示可应用于文本的各条命令。

如果在**新名称**框中双击标记名，则会出现它在 “标记名字典”中的定义。

如果清除该标记名，然后双击空白的**新名称**框，则会出现 “标记浏览器”。

转换占位符标记名

索引标记名（以移除其服务）或从当前应用程序导入 / 导出一个窗口或 QuickScript 时，所有与该窗口或 QuickScript 关联的标记名将随窗口一起移动，但它们不会添加到新应用程序的数据库中。相反，它们会自动标为“占位符”（索引）标记名。您必须转换这些占位符标记名，并且如有必要，请在新应用程序的“标记名字典”中定义它们。例如：



在本例中，要将占位符标记名转换为本地标记名，请单击**转换**。

提示 导入一个窗口时，如果任何一个标记名（远程标记名除外）尚未在本地“标记名字典”中定义，则系统会提示您在转换之前先去定义它们。如果确实是这种情形，请单击**确定**。此时会出现**标记名字典**对话框供您去定义这些标记名。

请注意标记名前面的占位符 **?d:**、**?i:**、**?m:** 及 **?r:**。它们指明了标记名原先定义的类型：

占位符符号	标记名类型
d	离散型
i	整型
m	消息型
r	实型

远程引用不会显示为占位符，而是显示为远程标记名引用，例如 **PLC2:Temperature**。

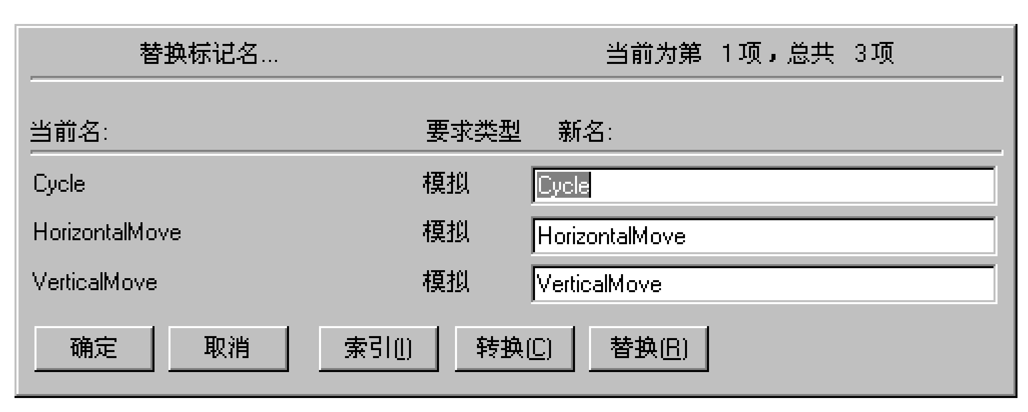
将标记名转换为远程引用

您可以在**替换标记名**对话框中使用多种方法，将占位符（或本地）标记名转换为远程标记名引用。您可以直接输入远程标记名引用；转换与导入窗口关联的占位符标记名；或者，打开“标记浏览器”并显示标记源的“标记名字典”以选择远程标记名引用。

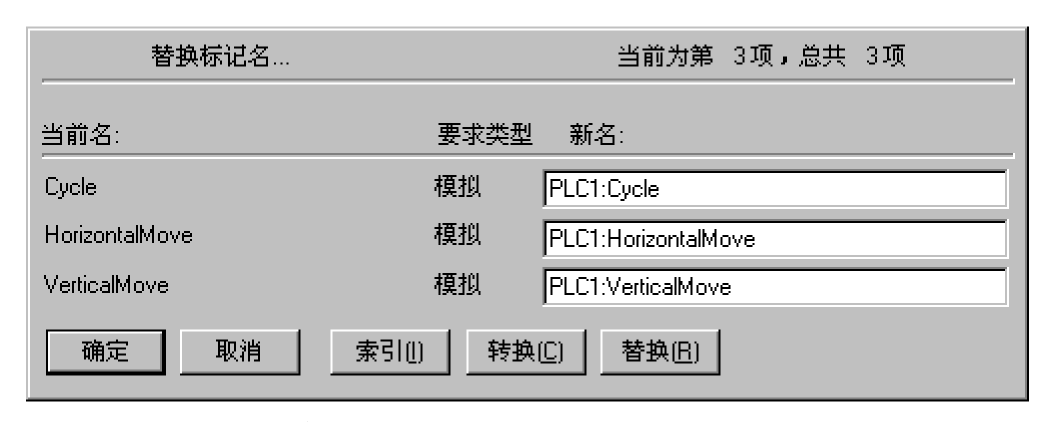
要手动将标记名转换为远程标记名引用

1. 选择与要转换为远程标记名引用的本地标记名关联的对象，然后在**特别**菜单上，单击**替代标记**。此时会出现**替换标记名**对话框。

> **提示** 要快速访问该对话框，请选择所有对象，然后右击所选对象之一，指向**替换**，然后单击**替换标记**。



2. 在**新名称**框中，选择要更改的每个标记名，然后输入远程标记名引用：



3. 单击**确定**。

提示 如果您使用这种方法并且不再需要在本地“标记名字典”中定义这些原始标记名，则可以更新标记名使用计数，然后删除未使用的标记名。

如需有关详细信息，请参阅“从字典中删除标记名”。

要将导入窗口的标记名转换为远程引用

1. 打开导入的窗口，选择所有对象，然后在**特别**菜单上，单击**替代**标记。此时会出现**替换标记名**对话框。

   如需有关导入窗口和脚本的详细信息，请参阅第 2 章“使用 WindowMaker”。

   提示 要快速访问该对话框，请按 F2 键（选择窗口中的所有对象），右击所选对象之一，然后指向**替换**，再单击**替代标记**。

替换标记名... 当前为第 1 项，总共 6 项

当前名:	要求类型	新名:
?d:Auto	离散	?d:Auto
?d:Ejector	离散	?d:Ejector
?i:Cycle	模拟	?i:Cycle
?r:HorizontalMove	模拟	?r:HorizontalMove
?r:Speed	模拟	?r:Speed
?r:VerticalMove	模拟	?r:VerticalMove

确定 取消 索引(I) 转换(C) 替换(R)

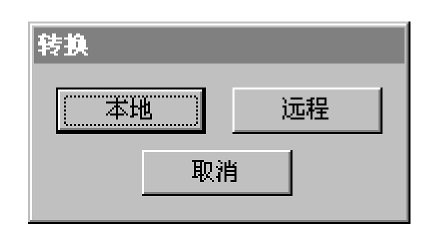
备注 **索引**按钮可以将链接转换为占位符，使动画链接不能指向本地标记名，从而使之不会被意外删除。

您也可以使用**替代标记**命令将本地标记名转换为远程标记名引用。为此，请选择与本地标记名关联的对象，然后选择**替代标记**命令，以便在**替换标记名**对话框中显示它们。单击**索引**将它们转换为定位符标记名，然后单击**转换**并按下述步骤继续操作。

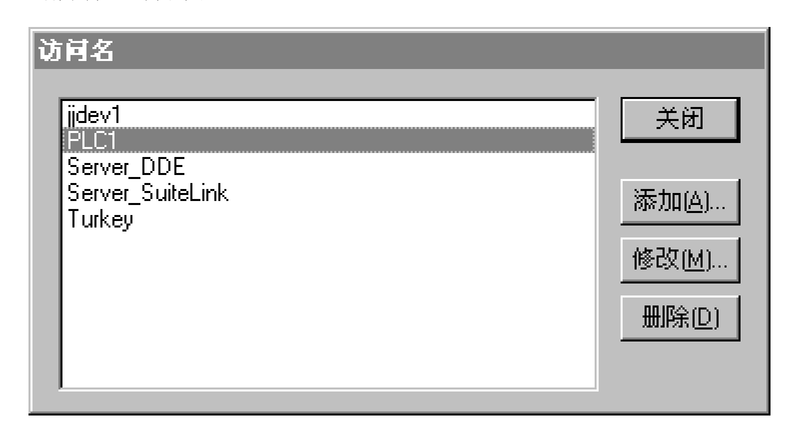
如果在上述任一选项中使用**索引**，而且不再需要在本地“标记名字典”中定义这些原始标记名，则可以更新标记名使用计数，然后删除未使用的标记名。

如需有关详细信息，请参阅“从字典中删除标记名”。

2. 单击**转换**。此时会出现**转换**对话框。



3. 单击**远程**。此时会出现**访问名**对话框，列出在本地应用程序中定义的所有 “访问名”。



4. 在列表中双击 “访问名”，或选择它然后单击**关闭**。

提示 要验证 “访问名”是否正确，请单击**修改**以查看它。

如果当前尚未定义指向该标记源的 “访问名”，请单击**添加**并立即去定义它。“访问名”<u>必须</u>包含应用程序所在的远程节点。

---

5. <u>替代标记</u>对话框中显示的**所有**标记名将自动转换为远程标记名引用（将所选 “访问名”用作前缀）。例如：



6. 单击**确定**。

提示 此操作程序对于导入的 QuickScript 相同有效，不同的是要在 InTouch QuickScript 编辑器中打开 QuickScript，然后单击**转换**。

通过从另一个应用程序导入窗口或 QuickScript，并将所有与动画链接或 QuickScript 关联的标记名转换为远程标记名引用，您不必在本地“标记名字典”定义单个标记名，就可以立刻接收来自数以百计的远程标记名的数据。

要在标记浏览器中选择远程标记名引用

1. 选择与要转换为远程标记名引用的本地标记名关联的对象，然后在**特别**菜单上，单击**替代标记**。此时会出现**替换标记名**对话框。

提示 要快速访问该对话框，请选择所有对象，然后右击所选对象之一，指向**替换**，然后单击**替换标记**。

---

2. 清除**新名称**框中要用远程标记名引用取代的标记名，然后双击**新名称**框。此时会出现**选择标记**对话框。

3. 单击 工具以显示树形视图窗格：



4. 如果已在“标记浏览器”中定义标记源，请在树形视图中选择它以显示其标记名。

提示 要展开树形视图，请双击应用程序名或单击 ⊞。

备注 如果当前尚未定义标记源，您现在便可以去定义它。

如需有关定义标记浏览源的详细信息，请参阅 “定义标记源”。

5. 双击要使用的远程标记名，或选择它然后单击**确定**。

提示 您也可以选择 SuperTag 成员标记名。例如：



此时会出现**替代标记**对话框，显示着远程标记名引用：



6. 单击**确定**关闭对话框，并将远程标记名与所选对象关联起来。

> **提示** 对于要在**替换标记名**对话框中改变的每个标记名，重复此操作程序。

要替换标记名

1. 选择要更改其标记名的所有对象。
2. 在**特别**菜单上，单击**替代标记**。此时会出现**替换标记名**对话框。

> **提示** 要快速访问该对话框，请右击所选对象之一，指向**替换**，然后单击**替代标记**。



3. 单击**替换**。此时会出现**替换文本**对话框。



4. 在**旧文本**框中，输入要替换的标记名部分。
5. 在**新文本**框中，输入替换文本。

6. 单击**确定**。此时**替换标记名**对话框会再次出现，显示对该标记名所作的更改：

替换标记名... 当前为第 1 项，总共 6 项

当前名:	要求类型	新名:
Auto	离散	Auto
Cycle	模拟	Cycle
Ejector	离散	Ejector
HorizontalMove	模拟	HorizontalPos
Speed	模拟	Speed
VerticalMove	模拟	VerticalPos

确定 取消 索引(I) 转换(C) 替换(R)

7. 单击**确定**。包含已替换文本的所选对象的所有标记名会自动更改。

定标 I/O 标记名

所有 I/O 型标记名都从另一个 Windows 应用程序 （如 Excel 和 I/O 服务器）接收它们的值。这个值被称作 “原始”值。在 “标记名字典”中定义标记名时，您必须输入 “最小原始数据”和 “最大原始数据”值。数据库使用这些值来限制从 I/O 设备获取的实际原始值。例如，如果将 “最小原始数据”值设置为 50，但从 I/O 服务器收到的实际值是 0，则数据库会将原始值强行设置为 50。

InTouch 不显示原始值。相反，它会显示工程单位 (EU)。在 “标记名字典”中定义 I/O 型标记名时，您必须指定 “最小工程单位”和 “最大工程单位”值。这些值用于将原始值定标为显示值。如果不需要定标或 I/O 设备已执行定标，请将 “最小 / 最大工程单位”值设置为等于 “最小 / 最大原始数据”值。

例如，假设有一个连接到 PLC 寄存器的流量发送器，在无流量时它值为零，在流量为 100% 时它的值为 9999。您会输入下面的各个值：

最小工程单位 = 0 最大工程单位 = 100

最小原始数据 = 0 最大原始数据 = 9999

原始值 5000 将显示为 50。

我们再假设有一个连接到 PLC 寄存器的流量发送器，无流量时它的值是 6400，流量为 300GPM 时它的值为 32000。

最小工程单位 = 0 最大工程单位 = 300

最小工程单位 = 6400 最大工程单位 = 32000

在这种情况下，原始值 12800 将显示为 150。原始值 6400 将显示为 0，原始值 0 将显示为 0 （超出 “最小原始数据” 和 “最大原始数据” 值所设定的边界的所有值均会被调整）。

I/O 型标记名数据从 InTouch “标记名字典” 写入其它 Windows 应用程序时，上述定标操作相反。

例如，操作员可能会在数据输入窗口中输入设定点值 0-300 GPM，而系统则将 6400-32000 的值传输给 PLC 寄存器。

在 InTouch 程序充当客户端 （或者是要求提供数据，或者是发送数据给另一个 Windows 程序），这总是有效的。不过，在 InTouch 充当数据源时 （另一个 Windows 程序要求 InTouch 提供数据），则它会将 “工程单位” 值返回给发出请求的程序。

例如，如果在 Excel 电子表格中的某个单元格包含远程引用公式：

```
=view|tagname!speed
```

单元格中显示的值会是标记名 speed 的当前 EU 值。

仪器故障监视

从 7.0 版开始，InTouch 支持三个标记名**点域**（**.RawValue**、**.MinRaw** 和 **.MaxRaw**），供您在 InTouch QuickScripts 中用来监视仪器的数值，以确定是否超出范围、需要校准或出现故障。

内部系统 $Tagnames

InTouch 提供了大量预定义的内部系统标记名，供您用于执行各种操作。例如，如果要显示当前的时间，则可以将系统标记名 **$Timestring** 链接到一个值显示字符串。所有的内部标记名均以美元号 (**$**) 开头。内部系统标记名可通过 “标记浏览器” 进行访问。

如需有关详细信息，请参阅 “标记浏览器”。

下节简要介绍内部系统标记名：

系统标记名	描述
$AccessLevel	只读整型安全标记名，在表达式或脚本中用以控制操作员执行特定功能的能力。
$AlarmLogging	在 InTouch 中不再受支持。
$AlarmPrinterError	在 InTouch 中不再受支持。
$AlarmPrinterNoPaper	在 InTouch 中不再受支持。
$AlarmPrinterOffline	在 InTouch 中不再受支持。
$AlarmPrinterOverflow	在 InTouch 中不再受支持。

系统标记名	描述
$ApplicationChanged	只读离散型标记名，反映分布式系统中远程应用程序是否发生改变。每次在服务器节点的 WindowViewer **特别**菜单上选择**通知客户**命令时，此数值都会递增。
$ApplicationVersion	只读实型标记名，反映应用程序的当前版本号。每次更改、添加或删除标记名或 QuickScript 时，此数值均会改变。
$ChangePassword	只写离散型安全标记名，允许操作人员将 **$ChangePassword** 标记名的值设置为 1，从而给操作员显示一个普通的 “改变口令” 对话框。
$ConfigureUsers	只写离散型安全标记名，可用在离散型按钮上，供操作员将 **$ConfigureUsers** 标记名的值设置为 1，从而显示一个普通的**配置用户**对话框以便编辑安全用户名列表。
$Date	只读整型标记名，显示从 1970 年 1 月 1 日起的天数之整数部分。
$DateString	只读内存消息型标记名，以 WIN.INI 文件中设置的相同格式显示日期，例如 4/18/1992。（此日期格式可通过 Windows “控制面板” 进行设置）。
$DateTime	只读实型标记名，显示从 1970 年 1 月 1 日起的天数之小数部分。
$Day	只读整型标记名，显示当前日期（值可以 1 到 31）。
$HistoricalLogging	可读 / 写离散型标记名，用于监视或控制历史记录功能的启动和停止。对于整个应用程序而言，这是一个全局命令。
$Hour	只读整型标记名，显示当前的小时值（值可以是 0 到 23）。
$InactivityTimeout	只读离散型安全标记名，配置的操作员自动注销时间已过时，它会被设置为 1。
$InactivityWarning	只读离散型安全标记名，配置的警告操作人员注销即将发生的时间已过时，它的值会被设置为 1。

系统标记名	描述
$LogicRunning	可读 / 写离散型标记名，用于监视或控制脚本的运行。 当前正在执行的异步用户自定义函数脚本无法停止。不过，您可以防止去执行新的脚本。
$Minute	只读整型标记名，显示当前的分钟值（值可以是 0 到 59）。
$Month	只读整型标记名，显示当前的月份（值可以是 1 到 12）。
$Msec	只读整型标记，显示当前的毫秒数（值可以是 0 到 999）。
$NewAlarm	可读写离散型标记名，每当发生新的报警时，它的值便会被设置为 1。
$ObjHor	只读整型标记名，用于显示所选对象中点的水平像素位置。
$ObjVer	只读整型标记名，用于显示所选对象中点的垂直像素位置。
$Operator	只读安全消息型标记名，可在表达式或 QuickScript 中用以控制操作人员执行特定功能的能力。
$OperatorEntered	可读写安全消息型标记名，用于设置操作员的“用户名”。
$PasswordEntered	只写安全消息型标记名，用于设置操作员的“口令”。
$Second	只读整型标记名，显示当前的秒数（值可以是 0 到 59）。
$StartDdeConversations	可读 / 写离散型标记名，用于在**特别**菜单被禁用时启动未初始化的对话。
$System	只读“报警组”型标记名，用作根报警组。如果未给特定的 " 报警组 " 名指定标记名，则缺省条件下它会自动指定给这个根组。定义的所有“报警组”都是 **$System** 的后代。
$Time	只读整型标记名，用毫秒显示自午夜以来的时间。

系统标记名	描述
$TimeString	只读内存消息型标记名，按 Windows 注册表中设置的格式显示时间，如 12:01:59 PM。（此格式可以通过 Windows “控制面板”进行设置。）
$Year	只读整型标记名，用 4 位数字显示年份。例如 2008。

如需有关系统标记名的详细信息，请参阅联机 *InTouch 参考指南*。

标记名点域

大多数关于 InTouch 的讨论都涉及到对象的概念。对象的概念很广泛也很复杂。在这里，我们将对象的定义限制为关于屏幕上的一个图形对象的信息集合，或者是关于 “标记名字典”中的一个标记名的信息集合。

例如，如果在屏幕上绘制一个矩形，此矩形会有一些特定的 “属性”，如线宽、线颜色、填充颜色、屏幕上的像素位置、关联的链接，等等。标记名与此非常相似。例如，创建一个模拟型报警标记名 "Analog_Tagname"，则它会具有一些关联的属性，如标记名的名称、标记名的 **HiHi** 报警设定点。等等。这些属性中有一些可以在 InTouch 中通过脚本、表达式以及用户输入进行访问，它们也就是所谓的**点域**。访问与标记名关联的这些数据域时需要使用的语法是：**标记名点域**。

例如，为了允许在运行时改变标记名 "Analog_Tagname" 上的 **HiHi** 报警限，您可以创建一个**模拟 - 用户输入**触动链接，并将它应用到一个按钮，该按钮是使用表达式 **Analog_Tagname.HiHiLimit** 定义的。在运行时，操作员可以单击该按钮，并且为 "Analog_Tagname" 使用的 **HiHi** 报警限输入一个新值，处理方式非常简单。

您可以使用**点域**来允许输入和输出与标记名关联的数据，您还可以使用历史**点域**来允许用户动态修改显示的历史趋势。例如，您可以允许用户控制滚动、锁定或重新定位趋势图上的指示器，或重新将笔赋给新的标记名。

如需有关**点域**的详细信息，请参阅联机 *Intouch 参考指南*。

要访问标记名点域

在任意 InTouch QuickScript、动画链接标记名或表达式输入框中，输入标记名加一个句点 (**tagname.**)，然后双击句点 (**.**) 右侧。此时会出现 “选择标记” 浏览器，显示为当前标记源定义的标记名：



要选择点域

1. 单击**点域**箭头以打开**点域**列表，从中您可以将选择点域让它与当前所选的标记名类型关联。

提示 缺省条件下，所有类型的标记名最初均显示 < 无 >。

备注 从 “标记名字典” 访问 “标记浏览器”、在运行期间从 “历史趋势设置” 对话框为历史趋势笔选择标记名时，均不能使用**点域**。（配置历史趋势必须使用 “允许运行时改变” 选项进行配置）。

2. 在列表中单击要赋给所选标记名的**点域**。

备注 并非每个标记名类型都有相同的**点域**。例如，**离散**型标记名有 **.OnMessage** 点域，而模拟型则没有。如果选择**离散**型标记名并为它指定 **.OnMessage**，然后选择另一个**离散**型标记名，则显示的**点域**列表不会改变。但是，如果在控制视图列表中选择另一种类型的标记名 （如模拟型），则显示的**点域**将恢复为 < 无 >。

如需有关 “标记浏览器” 的详细信息，请参阅 “标记浏览器”。

下节将简要介绍标记名**点域**：

点域	描述
.Ack	可读 / 写离散型标记名点域，用于监视或控制标记名和 “报警组” 的报警确认状态。 **.Ack** 有一个逆向标记名点域 **.Unack**。出现未确认的报警时， **.Unack** 将设置为 1。 **.Unack** 随后可用在动画链接或条件脚本中，以针对未确认的报警触发电铃。
.AckDev	监视或控制标记名上活动偏差报警的报警确认状态。
.AckDsc	监视或控制报警或离散型标记的报警确认状态。
.AckROC	监视或控制标记名上活动变化率报警的报警确认状态。
.AckValue	监视或控制标记名上活动值报警的报警确认状态。
.Alarm	只读离散型点域，指定的标记名、“报警组名” 存在报警条件时会被设置为 1。
.AlarmAccess	返回与所选报警关联的标记名的访问名。报警必须通过单击摘要 “分布式报警显示” 进行选择。
.AlarmAckModel	监视与标记名关联的**确认模型**，具体如下： 0 = 条件 （缺省值） 1 = 面向事件 2 = 扩展的摘要 应用于含报警的离散或模拟型标记名。只读，但可以在 WindowMaker 中进行配置。
.AlarmClass	返回报警的类。
.AlarmComment	可读 / 写文本字符串，描述报警而非标记名的内容。缺省条件下，它在新的应用程序中为空。 不过，在将 InTouch 7.1 应用程序转换为 7.11 应用程序时，为保证后向兼容性，标记名注释会被复制到 **AlarmComment** 中。

点域	描述
.AlarmDate	返回报警日期。
.AlarmDev	发出存在偏差报警的信号。
.AlarmDevCount	记录给定标记名或报警组上活动偏差报警的总数。
.AlarmDevDeadband	可读 / 写模拟 （只对整型或实型标记名有效）型标记名点域，用于监视或控制副偏差报警和主偏差报警的偏差死区百分比。
.AlarmDevUnAckCount	记录给定标记名或报警组上的未确认活动偏差报警总数。
.AlarmDisabled	禁用或启用事件和报警。应用于含报警的离散或模拟型标记名，或应用于报警组。 **备注**此点域与 **.AlarmEnabled** 相同，但极性相反。
.AlarmDsc	指出离散报警条件当前处于活动状态。
.AlarmDscCount	记录给定标记名或报警组上活动离散报警的总数。
.AlarmDscDisabled	指出标记名是否可产生离散报警。 **备注**对于离散型标记名，此点域与 **.AlarmDisabled** 相同。
.AlarmDscEnabled	指出标记名是否可生成离散报警。 **备注**对于离散型标记名，此点域与 **.**AlarmEnabled 相同。
.AlarmDscInhibitor	返回指定给此标记名的离散报警的约束标记名 （如果有）。
.AlarmDscUnAckCount	记录给定标记名或报警组上活动的未确认离散报警总数。
.AlarmEnabled	可读 / 写离散型标记名点域，用于停用或启用标记名和 “报警组”的事件和报警。
.AlarmGroup	包含用于填写分布式报警显示对象的当前查询。
.AlarmGroupSel	返回报警的报警组。
AlarmHiDisabled	停用或启用含报警的模拟型标记名的 **Hi** 报警限。

点域	描述
.AlarmHiEnabled	停用或启用含报警的模拟型标记名的 **Hi** 报警限。 **备注**与对应的停用点域相同，但极性相反。
.AlarmHiInhibitor	返回 **Hi** 报警限的**报警约束标记**引用。应用于含报警的离散型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmHiHiDisabled	停用或启用含报警的模拟型标记名的 **HiHi** 报警限。
.AlarmHiHiEnabled	停用或启用含报警的模拟型标记名的 **HiHi** 报警限。 **备注**与对应的停用点域相同，但极性相反。
.AlarmHiHiInhibitor	返回 **HiHi** 报警限的**报警约束标记**引用。应用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmLimit	返回报警的报警限。
.AlarmLoDisabled	停用或启用含报警的模拟型标记名的 **Lo** 报警限。
.AlarmLoEnabled	启用或停用含报警的模拟型标记名的 **Lo** 报警限。 **备注**与对应的停用点域相同，但极性相反。
.AlarmLoInhibitor	返回 **Lo** 报警限的**报警约束标记**引用。应用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmLoLoDisabled	停用或启用含报警的模拟型标记名的 **LoLo** 报警限。
.AlarmLoLoEnabled	停用或启用含报警的模拟型标记名的 **LoLo** 报警限。 **备注**与对应的停用点域相同，但极性相反。

点域	描述
.AlarmLoLoInhibitor	返回 **LoLo** 报警限的**报警约束标记**引用。应用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmMajDevDisabled	停用或启用含报警的模拟型标记名的**主偏差**报警限。
.AlarmMajDevEnabled	停用或启用含报警的模拟型标记名的**主偏差**报警限。 **备注**与对应的停用点域相同，但极性相反。
.AlarmMajDevInhibitor	返回**主偏差**报警限的**报警约束标记**引用。应用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmMinDevDisabled	停用或启用含报警的模拟型标记名的**副偏差**报警限。
.AlarmMinDevEnabled	停用或启用含报警的模拟型标记名的**副偏差**报警限。 **备注**与对应的停用点域相同，但极性相反。
.AlarmMinDevInhibitor	返回**副偏差**报警限的**报警约束标记**引用。应用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmName	返回报警的名称。
.AlarmOprName	返回与报警关联的操作员姓名。
.AlarmOprNode	返回与报警关联的操作员节点。
.AlarmPri	返回报警的优先级值 (1-999)。
.AlarmProv	返回报警的供应器。
AlarmROC	发出存在 “变化率”型报警的信号。
.AlarmROCCount	记录给定标记名或报警组上活动变化率报警的总数。
.AlarmROCDisabled	停用或启用含报警的模拟型标记名的**变化率**报警限。

点域	描述
.AlarmROCEnabled	停用或启用含报警的模拟型标记名的**变化率**报警限。 **备注**与对应的停用点域相同，但极性相反。
.AlarmROCInhibitor	返回**变化率**报警限的约束标记名引用。应用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.Alarm ROCUnAckCount	记录给定标记名或报警组上未确认的变化率报警总数。
.AlarmState	返回报警状态。
.AlarmTime	返回报警时间。
.AlarmTotalCount	记录给定标记名或报警组上活动报警的总数。
.AlarmType	返回报警类型。
AlarmUnAckCount	记录给定标记名或报警组上未确认的活动报警总数。
.AlarmUserDefNum1	可读 / 写实型 （浮点）值，缺省值为 0，且值未设置。应用于含报警的离散型标记名、含报警的模拟型标记名，或报警组。 **备注**此点域的值会附加在报警上，但 "只" 限于已设置 （例如通过脚本或插入）值的情况。
.AlarmUserDefNum2	可读 / 写实型 （浮点）值，缺省值为 0，且值未设置。应用于含报警的离散型标记名、含报警的模拟型标记名，或报警组。 **备注**此点域的值会附加在报警上，但 "只" 限于已设置 （例如通过脚本或插入）值的情况。
.AlarmUserDefStr	可读 / 写文本字符串，缺省值为 ""，且未设置值。应用于含报警的离散型标记名、含报警的模拟型标记名，或报警组。 **备注**此点域的值会附加在报警上，但 "只" 限于已设置 （例如通过脚本或插入）值的情况。

备注 .AlarmUserDefNum1、 **.AlarmUserDefNum2** 和 **AlarmUserDefStr** 点域可供您在报告时指定一个或多个要追加到报警记录的值。这些值由 Alarm DB Logger 写入数据库。您可以在报警上附加三个项目：两个数字和一个字符串。缺省条件下，它们为空 （零和 ""）。

要简化设置用户值，您可以在报警组及特定的标记名上设置它们。例如， InBatch 可以在 **$System** 报警组上设置 **.AlarmUserDefNum1** 中的批号，从而让所有的报警均附加上该批号。

如果在某个 "报警组" 上设置 **.AlarmUserDefNum1**，则它将应用于该组及其子组中的所有报警。您也可以在标记名上专门设置 **.AlarmUserDefNum1** 的值。在这种情况下，它只应用于该标记名，并且会改写标记名的 "报警组" 中 **.AlarmUserDefNum1** 的任何设置。

.

点域	描述
AlarmValDeadband	可读 / 写模拟型 （只对整型或实型标记名有效）点域，用于监视和 （或）控制报警死区的值。此点域对于 "报警组" 以及普通的标记名而言是有效的。
.AlarmValue	返回报警值。
.AlarmValueCount	记录给定标记名或报警组上活动值报警的总数。
.AlarmValueUnAckCount	记录给定标记名或报警组上未确认的活动值报警的总数。
.ChartLength	可读 / 写整型标记名 点域，用于控制显示在 "历史趋势" 图上的时间长度。 **.ChartLength** 以秒为单位显示图表长度。
.ChartStart	可读 / 写整型标记名点域，用于控制开始时间和 （或）滚动相应的历史趋势图表。 **.ChartStart** 显示自 1970 年 1 月 1 日午夜 12 点以来的时间 （以秒为单位）。
.Comment	只读消息型标记名点域，用于显示在 "标记名字典" 中为标记名输入的注释域。
.DevTarget	可读 / 写模拟型 （只对整型或实型标记名有效）点域，用于监视和 （或）控制副偏差和主偏差报警的目标。
.DisplayMode	可读 / 写模拟型标记名点域，用于确定在趋势图上显示值的方式。

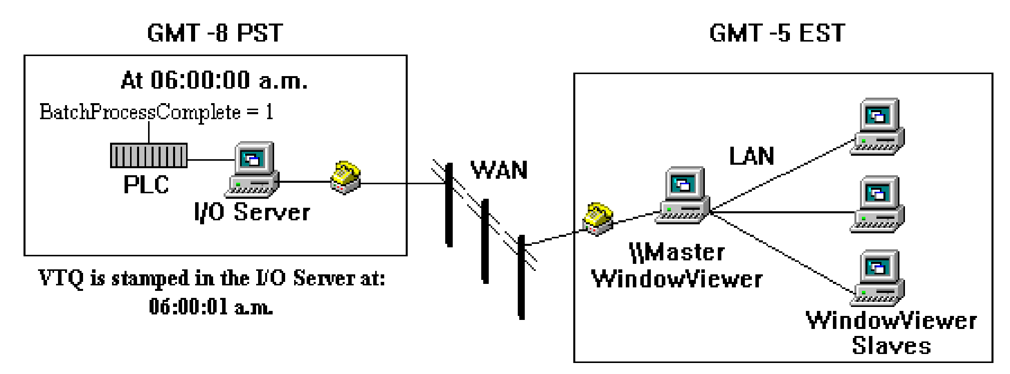
点域	描述
.EngUnits	可读 / 写模拟型标记名点域，用于访问在“标记名字典”中给模拟型标记名指定的工程单位。 **备注**写入的这些值不会保留。
.Freeze	读 / 写 “分布式报警显示对象”的冻结状态。
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	可读 / 写模拟型标记名点域，用于监视或控制值报警检查的极限值。这些点域只整型和实型标记名有效。
.HiStatus, .HiHiStatus, .LoStatus, .LoLoStatus	只读离散标记名点域，确定是否存在指定类型的报警。
.ListChanged	指出是否存在任何新报警或 “分布式报警对象”的更新。
.MajorDevPct	可读 / 写实型标记名点域，用于监视或控制报警检查的主偏差百分比。
.MajorDevStatus	只读离散型标记名点域，确定指定的标记名是否存在主偏差报警。
.MaxEU, .MinEU	只读整性标记名点域，显示标记名的最大值和最小值。
.MaxRange, .MinRange	可读 / 写实型标记名点域，用于表示当前绘制趋势的每个标记名所显示的标记名“工程单位”范围的百分比。 **.MaxRange** 和 **.MinRange** 范围限于从 **0** 到 **100**，并且 **.MinRange** 总是应小于 **.MaxRange**。如果这两个域中的一个被赋予的值小于 **0** 或大于 **100**，则这些值会被调整为 **0** 或 **100**。如果 **.MinRange** 大于或等于 **.MaxRange**，则趋势不会显示任何数据。
.MaxRaw, .MinRaw	WindowViewer 作为客户端从 I/O 服务器收到的实际原始数据的高 / 低嵌位设置。标记名点域 **.MaxRaw/.MinRaw** 的值来自标记名数据库中指定的 I/O 型标记名的 “最大 / 最小原始数据”值。超过或低于此设置的任何原始值均会调整为此值。

点域	描述
.MinorDevPct	可读 / 写实型标记名点域，用于监视或控制报警检查的副偏差百分比。
.MinorDevStatus	只读离散型标记名点域，用于确定特定标记名的副偏差报警是否存在。
.Name	可读 / 写消息型标记名点域，用于显示标记名的实际名称。例如，标记名 **TagID** 的名称。
.Normal	只读离散型标记名点域，指定的名称没有报警时，它的值会被设置为 1。此点域对于 “报警组” 以及普通的标记名而言是有效的。
.OffMsg, .OnMsg	可读 / 写消息型标记名点域，用于显示在 “标记名字典” 中为离散型标记名指定的开消息和关消息。 **备注**写入的这些值不会保留。
Pen1 - .Pen8	可读 / 写 **TagID** 型的标记名点域，用于控制每支笔当前正在绘制其历史趋势的标记名。 **备注** 将标记名赋给 “笔” 更简便方法是使用 “历史” 函数 **HTSetPenName** 和 **HTGetPenName**。 如需有关详细信息，请参阅第 12 章 “实时趋势和历史趋势”。 如需有关 TagID 标记名类型的详细信息，请参阅 “标记名类型”。
.Quality	只读整型标记名，可供用户访问 I/O 服务器提供的 I/O 标记名质量。
.QualityLimit	只读整型标记名点域，用于显示在 I/O 连接有效时数据源提供的 I/O 值的质量限。
.QualityLimitString	只读消息型标记名点域，用于显示在 I/O 连接有效时，I/O 服务器提供的 I/O 值的质量限字符串。
.QualityStatus	只读整型标记名点域，用于显示在 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量状态。

点域	描述
.QualityStatusString	只读消息型标记名点域，用于显示在 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量状态字符串。
.QualitySubstatus	只读整型标记名点域，用于显示在 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量子状态。
.QualitySubstatusString	只读消息型标记名点域，用于显示在 I/O 连接有效时，由 I/O 服务器提供的 I/O 值的质量子状态字符串。 备注 如果 I/O 连接变为无效，则质量点域会自动复位成初始值零。此时 **.ReferenceComplete** 标帜也会被设置为零。
.RawValue	任何类型 （实型 / 离散型）的标记名点域，用于显示在 InTouch 应用定标之前，实际的离散或模拟型 I/O 值。
.Reference	可读 / 写标记名点域，与 I/O 型标记名一起使用，以动态改变标记名来源的地址。
.ReferenceComplete	离散型标记名点域，确认所请求的项目也同样反映在 **.Value** 域中。 如需有关 .Reference 点域的详细信息，请参阅 “动态地址引用 (DRA)”。
.ROCPct	可读 / 写标记名点域，用于监视和 （或）控制报警检查的变化率。
.ROCStatus	只读离散型标记名点域，用于确定指定的标记名是否存在 “变化率”报警。
.ScooterLockLeft	可读 / 写离散型标记名点域。此点域值为 TRUE 时，“右”指示器不能移到左指示器位置的左边。（0=FALSE、 1=TRUE）。
.ScooterLockRight	可读 / 写离散型标记名点域，此点域的值为 TRUE 时，“左”指示器不能移动到右指示器的右边。（0=FALSE、 1=TRUE）。
.ScooterPosLeft	可读 / 写实型标记名点域，代表左指示器的位置 （范围是从 0.0 到 1.0）。

点域	描述
.ScooterPosRight	可读 / 写实型标记名点域，代表右指示器的位置 （范围是从 0.0 到 1.0）。
.SuppressRetain	读 / 写分布式报警显示对象的抑制保持状态。
.TagID	只读 **TagID** 型标记名点域，与 “历史趋势” **.Pen1-.Pen8 TagID** 标记名一起使用，以监视和 （或）控制某支笔当前绘制其趋势的标记名 （详见前文 "Pen" 点域的描述）。

下面显示 **.Time** 域如何接收其数据：



点域	描述
.TimeDate	实整型标记名点域，用于显示在 I/O 连接有效时，自 I/O 服务器提供 I/O 值以来已过的整数天数。
.TimeDateString	只读消息型标记名点域，使用 Windows 注册表中设置的相同格式来显示日期。例如，在 I/O 连接有效时， I/O 服务器提供 I/O 值的日期 10/31/1997。
.TimeDateTime	只读实型标记名点域，用于显示在 I/O 连接有效时，自 I/O 服务器提供 I/O 值以来已过的天数之小数部分。
.TimeDay	只读整型标记名点域，用于显示在 I/O 连接有效时， I/O 服务器提供 I/O 值的日期。（可以是 1 到 31）。
.TimeHour	只读整型标记名点域，用于显示在 I/O 连接有效时，I/O 服务器提供 I/O 值时的小时值。（值可以是 0 到 23）。
.TimeMinute	只读整型标记名点域，用于显示在 I/O 连接有效时，I/O 服务器提供 I/O 值时的分钟值。（值可以是 0 到 59）。
.TimeMonth	只读整型标记名点域，用于显示在 I/O 连接有效时， I/O 服务器提供 I/O 值时的月份。（值可以是 1 到 12）。
.TimeMsec	只读整型标记名点域，用于显示在 I/O 连接有效时， I/O 服务器提供 I/O 值的时间（以毫秒为单位）。
.TimeSecond	只读整型标记名点域，用于显示在 I/O 连接有效时， I/O 服务器提供 I/O 值的时间（以秒为单位）（值可以是 0 到 59）。
.TimeTime	只读整型标记名点型，用于显示在 I/O 连接有效时，I/O 服务器提供 I/O 值时的时间（自午夜算起，以毫秒为单位）。
.TimeTimeString	只读消息型标记名点域，用于显示在 I/O 连接有效时，I/O 服务器提供 I/O 值的日期和时间。

点域	描述
.TimeYear	只读整型标记名点域，使用同 Windows 注册表中设置的相同格式来显示时间。例如，在 I/O 连接有效时， I/O 服务器提供 I/O 值的时间 10/31/1997。（此格式可以通过 Windows “控制面板”进行设置。）
.Unack	可读 / 写离散型标记名点域，用于控制本地报警的报警未确认状态。
.UpdateCount	只读整型标记名点域，趋势检索完毕时它会递增。
.UpdateInProgress	只读离散型标记名点域，显示历史数据检索状态（0 = 未进行中的检索， 1 = 检索正在进行）。
.UpdateTrend	只写离散型标记名点域，设置为 1 时，会使用所有的当前值来更新 “历史趋势”图表。“历史趋势”自己不会自动更新。要更新图表并显示指定标记名的当前值，必须更改图表开始值、图表长度等。在按钮 QuickScript 中使用此点域，操作员将可以在运行时随时更新图表。
.Value	只读或可读 / 写模拟型标记名点域，显示指定标记名的值。

有多个报警**点域**与报警对象关联，而与标记名无关。

.

AlarmClass	**.AlarmAccess**	**.AlarmDate**
.AlarmGroupSel	**.AlarmLimit**	**.AlarmName**
.AlarmOprName	**.AlarmOprNode**	**.AlarmPri**
.AlarmProv	**.AlarmState**	**.AlarmTime**
.AlarmType	**.AlarmValue**	**.Freeze**
.ListChanged	**.NumAlarms**	**.PendingUpdates**
.SuppressRetain		

这些报警**点域**通过使用下面的函数进行访问：

GetPropertyM(ControlName.Property, MsgTag)。

运行期间执行时，从 **MsgTag** 获取属性，以便在选定的行中显示标记名。如果选择了多个行，则 **MsgTag** 的属性是其中第一行中的标记名。

如需有关报警**点域**的详细信息，请参阅联机 *Intouch 参考指南*。

模拟型标记名的地址位域

整型标记名中的单个位可以用位点域来编址。这些位被视为离散值 (0/1)，如果被写入，则模拟型标记名会立即更新。您可以在任何使用离散型标记名的地方使用位点域。例如，在 I/O、脚本、表达式等内容中。

位点域	描述
.00	第零位
.01	第一位
.02	以此类推
.	
.	
.	
.31	32 位整数中的最重要的位

下面是如何在表达式中使用位域的一个示例：

```
Temperature.08 == 1;
```

下面是如何在 QuickScript 中使用位域的一个示例：

```
IF Temperature.29 THEN
   Temperature.29 =0;

ENDIF;
```

标记名字典实用程序

InTouch 提供了两个 “标记名字典”实用程序：DBDump 和 DBLoad。DBDump 用于将 InTouch 应用程序 “标记名字典”作为文本文件导出，以便在另一个程序 （如 Microsoft Excel）中进行查看或编辑。 DBLoad 可供将采用适当格式的 “标记名字典”文件 （在另一个程序如 Excel 中创建的，或是从另一个 InTouch 应用程序中导出的 DBDump 文件）加载到现有的 InTouch 应用程序中。这两个程序可供单独复制、修改或开发数据库 （“标记名字典”）的各个部分，然后将它们合并到一个应用程序中。

DBLoad 实用程序也可用作 InTouch TemplateMaker 的替代方式，用于创建 SuperTag 实例。

如需有关详细信息，请参阅 “创建 SuperTag 实例”。

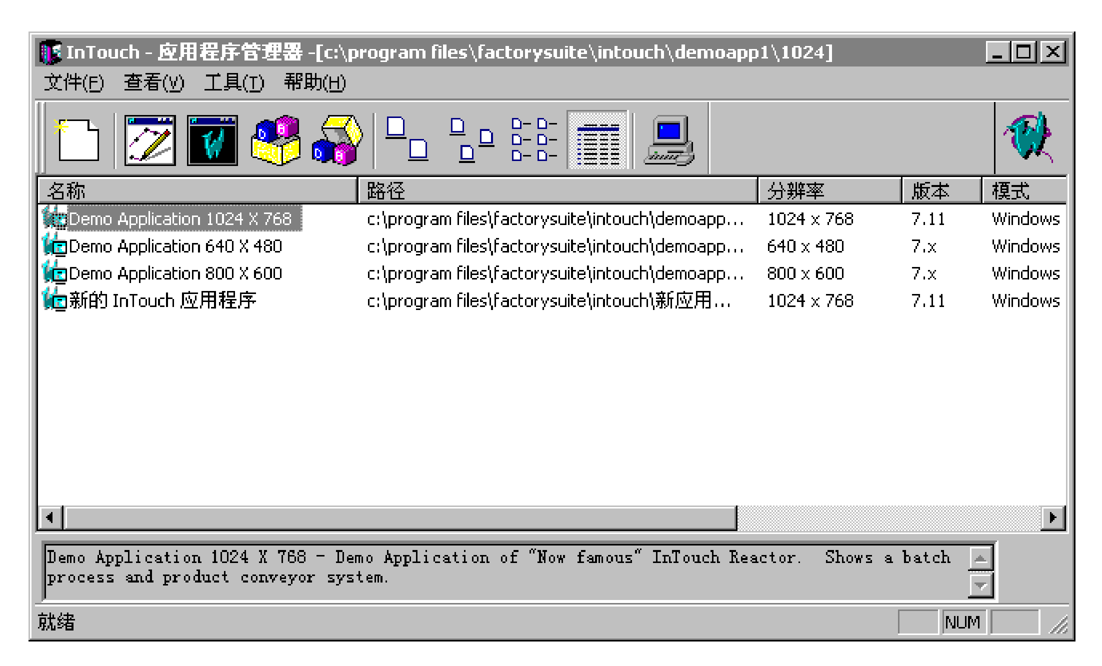
备注 DBDump 和 DBLoad 实用程序都是从 InTouch “应用程序管理器” (INTOUCH.EXE) 启动的。此外，您必须对使用较早版本的 InTouch 创建的应用程序进行转换，然后才能提取其 “标记名字典”。

如需有关创建数据库文件的详细信息，请参阅 “创建数据库输入文件”。

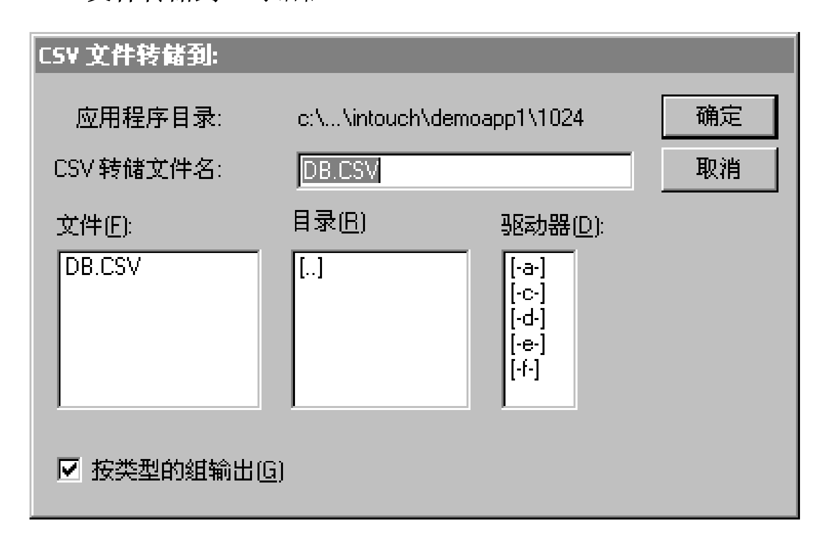
DBDump 实用程序

要提取现有 InTouch 应用程序的标记名字典

1. 如果 WindowMaker 和 WindowViewer 正在运行，请关闭它们。
2. 启动 InTouch。此时会出现 **InTouch 应用程序管理器**对话框。



3. 在**文件**菜单上，单击 **DBDump**，或单击 DBDump 工具。此时会出现 **CSV 文件转储到：**对话框。



4. 在 **CSV 文件转储文件名**框中，输入以 .CSV （逗号分隔变量）扩展名结尾的文件名。（如果已存在同名文件，则会出现一个消息框）。

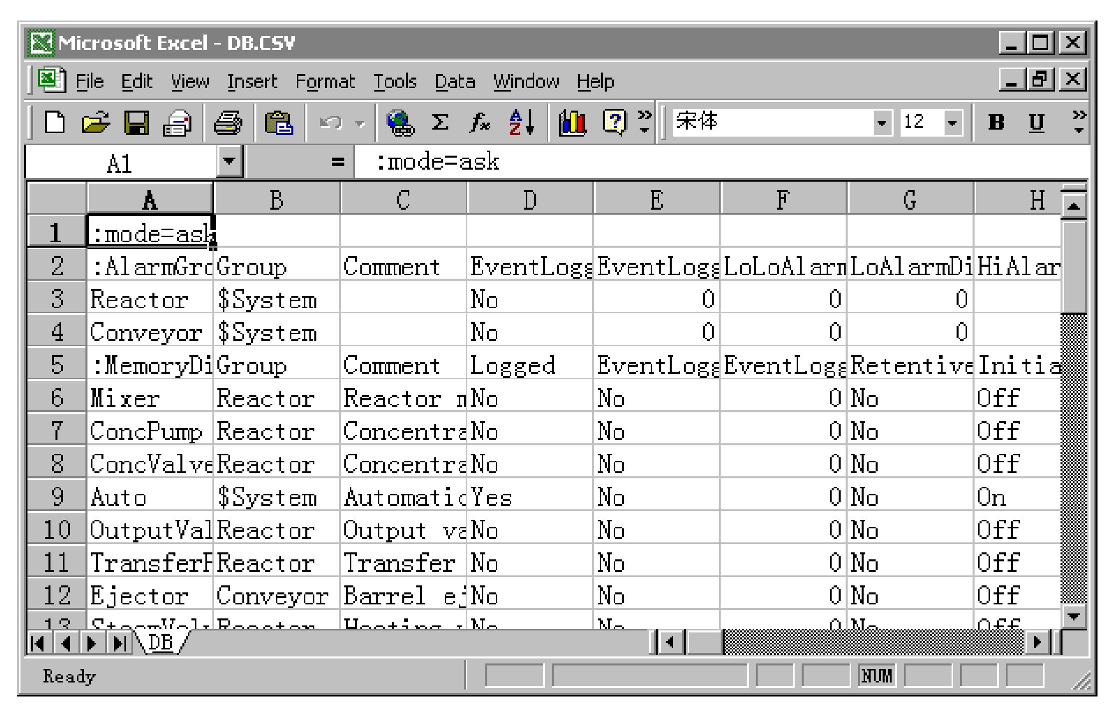
5. 选择**按类型**的组**输出**以按标记名类型而不是按标记名的字母顺序（缺省值）给提取的标记名分组。

备注 要按组提取数据库项目，系统必须针对每个标记名类型读取整个文件。因此，它需要更长时间来提取数据。不过，在支持 .CSV 的应用程序 （例如 Microsoft Excel）中打开输出文件时，分组会使读取或编辑更加容易。转储文件中包括每个标记名类型的占位符，而不管该类型的标记名是否存在。

---

6. 单击**确定**。所选应用程序目录下的数据库信息将下载到指定的文件名中。

如果在 Microsoft Excel 中打开 .CSV 文件，它会将逗号视作分隔符，并将数据记录自动分隔成列。例如：



	A	B	C	D	E	F	G	H
1	:mode=ask							
2	:AlarmGrc	Group	Comment	EventLogg	EventLogg	LoLoAlarm	LoAlarmDi	HiAlar
3	Reactor	$System		No	0	0	0	
4	Conveyor	$System		No	0	0	0	
5	:MemoryDi	Group	Comment	Logged	EventLogg	EventLogg	Retentive	Initia
6	Mixer	Reactor	Reactor m	No	No	0	No	Off
7	ConcPump	Reactor	Concentra	No	No	0	No	Off
8	ConcValve	Reactor	Concentra	No	No	0	No	Off
9	Auto	$System	Automatic	Yes	No	0	No	On
10	OutputVal	Reactor	Output va	No	No	0	No	Off
11	TransferF	Reactor	Transfer	No	No	0	No	Off
12	Ejector	Conveyor	Barrel e	No	No	0	No	Off

如果在 “记事本” 中打开 .CSV 文件，每条数据记录均由逗号隔开。例如：



```
:mode=ask
:AlarmGroup,Group,Comment,EventLogged,EventLoggingPriority,LoLoAlarmDisable,LoAlarm
"Reactor","$System","",No,0,0,0,0,0,0,0,0,0,"","","","","","","",""
"Conveyor","$System","",No,0,0,0,0,0,0,0,0,0,"","","","","","","",""
:MemoryDisc,Group,Comment,Logged,EventLogged,EventLoggingPriority,RetentiveValue,In
"Mixer","Reactor","Reactor mixer",No,No,0,No,Off,"Stopped","Started",None,1,"Reacto
"ConcPump","Reactor","Concentrate pump",No,No,0,No,Off,"Stopped","Started",None,1,"
"ConcValve","Reactor","Concentrate valve",No,No,0,No,Off,"Closed","Open",None,1,"Co
"Auto","$System","Automatic mode",Yes,No,0,No,On,"Manu","Auto",None,1,"Automatic mo
"OutputValve","Reactor","Output valve",No,No,0,No,Off,"Closed","Opened",None,1,"Out
"TransferPump","Reactor","Transfer pump",No,No,0,No,Off,"Stopped","Started",None,1,
"Ejector","Conveyor","Barrel ejector",No,No,0,No,Off,"Repos","Pouss?,None,1,"Barrel
"SteamValve","Reactor","Heating valve",No,No,0,No,Off,"Closed","Opened",None,1,"Hea
"WaterValve","Reactor","Water injection valve",No,No,0,No,Off,"Closed","Opened",Non
"TransferValve","Reactor","Transfer valve",No,No,0,No,Off,"Closed","Opened",None,1,
:IODisc,Group,Comment,Logged,EventLogged,EventLoggingPriority,RetentiveValue,Initia
:MemoryInt,Group,Comment,Logged,EventLogged,EventLoggingPriority,RetentiveValue,Ret
"SetPoint","Reactor","Concentrate setpoint",Yes,No,0,No,No,0,0,"",50,0,100,0,0,Off,
"Counter","$System","Program counter",No,No,0,No,No,0,0,"",0,0,999,0,0,Off,0,1,Off,
"Step1","Reactor","Batch process steps",Yes,No,0,No,No,0,0,"",0,0,10,0,0,Off,0,1,Of
"Cycle","Conveyor","Conveyor steps",No,No,0,No,No,0,0,"",0,0,999,0,0,Off,0,1,Off,0,
"BatchNumber","Reactor","Batch number in process",No,No,0,No,No,0,0,"",0,0,100,0,0,
:IOInt,Group,Comment,Logged,EventLogged,EventLoggingPriority,RetentiveValue,Retenti
```

DBLoad 实用程序

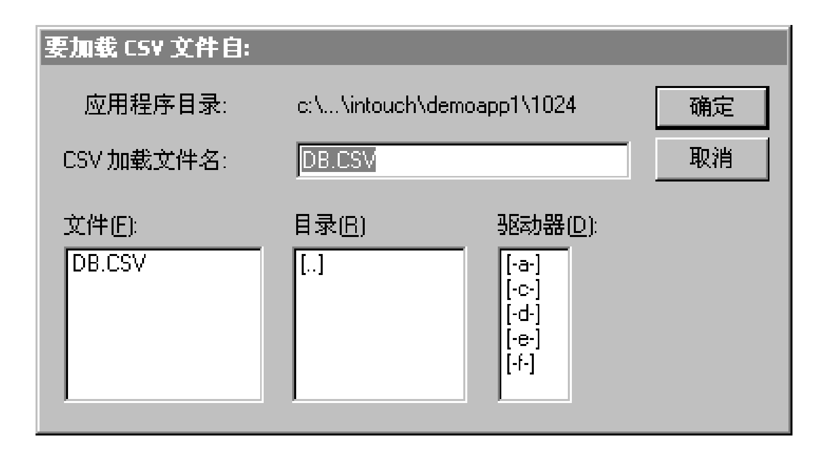
要加载 / 合并数据库输出文件至现有的 InTouch

1. 如果 WindowMaker 和 WindowViewer 正在运行，请关闭它们。

2. 启动 InTouch。此时会出现 **InTouch 应用程序管理器**对话框。



3. 在**文件**菜单上，单击 **DBLoad**，或单击 DBLoad 工具。此时会出现一个消息框，询问是否已备份应用程序。单击**是**继续。此时会出现 **CSV 文件 加载自：**对话框。



4. 在 **CSV 加载文件名**框中，输入要加载的 .CSV 文件的路径，或者使用**目录**和**驱动器**列表框找到文件。（正确选择文件之后，它的名称会出现在该框中）。

5. 单击**确定**。所选文件中包含的数据库信息将开始加载到所选应用程序的 “标记名字典” 中。

创建数据库输入文件

DBDump 和 DBLoad 数据库实用程序是用于在 “标记名字典” 上执行批处理类操作的工具。数据库输入文件可以在任何支持逗号分隔变量文件格式 (.CSV) 的程序中创建。（数据库输入文件必须以逗号分隔变量的格式来创建）。例如，“写字板”、“记事本”和 Microsoft Excel。创建输入文件之后，您就可以使用 DBLoad 程序将文件中包含的数据加载 / 合并到现有的 InTouch 应用程序数据库中。

您可以通过创建一个新的 InTouch 应用程序来创建数据库输入文件 “模板”，然后运行 DBDump 程序来转储其数据库，以创建一个格式正确的 .CSV 文件。如此一来，输入修改内容比从头创建输入文件要容易许多。

如需有关详细信息，请参阅 “创建数据库记录模板”。

数据库输入文件格式

数据库输入文件的第一行应指定文件的操作模式 **:mode**，通过 **DBLoad** 将文件加载 / 合并到应用程序时便会使用此模式。

提示 如果未指定模式 **:mode=test**，则缺省条件下将使用 **Ask**。

如需有关有效模式关键字的详细信息，请参阅 “数据库输入文件操作 模式”。

所有数据记录必须以标记名 **:type** 的有效关键字开头，后跟每条数据记录的有效关键字 （以逗号分隔）：

```
:mode=test
:IOMsg,Group,Comment,Logged,Event Logged,Event Logging
    Priority, . . .
```

每个标记名类型和数据记录都有一个有效的关键字

如需有关有效标记名类型和数据记录项的详细信息，请参阅 “:Type 和关键字 条目”。

随后输入实际的标记名，后跟每条数据记录的值 （用逗号分隔）：

```
:mode=test
IOMsg,Group,Comment,Logged,EventLogged,EventLoggingPriorit
    y,_ . .
Ingredient_1,$System,"",No,No,999, . . .
```

在上例中， **IOMsg** = Ingredient_1、 **Group** = $System、 **Logged** = No、 **EventLogged** = No、 **EventLoggingPriority** = 999。由于输入了 **""**，所以数据记录 **Comment** 会是空的。

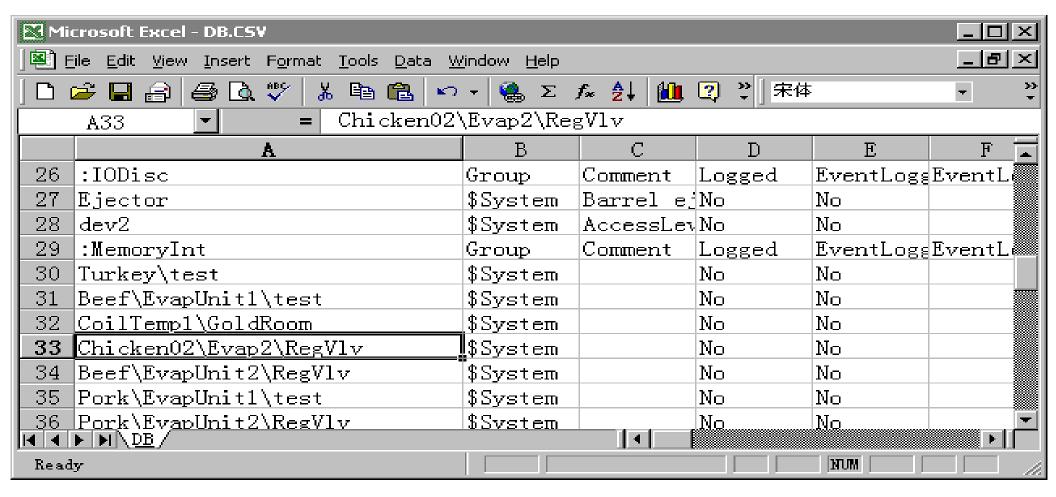
模式和类型关键字之前必须有一个冒号 (**:**)。如果要续行，请在行尾输入一个反斜杠 (\)。如果要输入注释，请在它前面加上一个分号 (**;**)。

创建 SuperTag 实例

除了 TemplateMaker、动画链接、 InTouch QuickScript 和 “标记名字典” 以外， InTouch 还支持通过 DBLoad 实用程序来创建 SuperTag。

备注 使用 DBLoad 创建 SuperTag 实例时，它们不会反映在 TemplateMaker 的 SuperTag 模板定义中。

通过 DBLoad 来创建 SuperTag 时，必须使用有效的 SuperTag 格式。SuperTag 实例数据记录必须以标记名 **:type** 的有效关键字开始。例如：



	A	B	C	D	E	F
26	:IODisc	Group	Comment	Logged	EventLogg	EventL
27	Ejector	$System	Barrel e	No	No	
28	dev2	$System	AccessLev	No	No	
29	:MemoryInt	Group	Comment	Logged	EventLogg	EventL
30	Turkey\test	$System		No	No	
31	Beef\EvapUnit1\test	$System		No	No	
32	CoilTemp1\GoldRoom	$System		No	No	
33	Chicken02\Evap2\RegVlv	$System		No	No	
34	Beef\EvapUnit2\RegVlv	$System		No	No	
35	Pork\EvapUnit1\test	$System		No	No	
36	Pork\EvapUnit2\RegVlv	$System		No	No	

下面的语法示例是有效的：

ParentInstance\ChildMember

ParentInstance\ChildMember\Submember

下面的语法示例是无效的：

ParentInstance

ParentInstance\ChildMember

如果使用的格式无效，则出现一个错误消息框，提醒您该语法无效。

加载包含 SuperTag 实例的 .csv 文件时，它们会自动添加到 “标记名字典” 中，并且可立即用于动画链接和 InTouch QuickScript 中。

空字符串与无输入值

空字符串字段与无输入值字段还是有些不同。例如：

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System
```

其中：

Tagname1 的注释字段为 **Hi**，而 Tagname2 的注释字段为空。 Excel 会如按上述内容读取 .CSV 文件，但保存文件时，它会按如下方式保存：

:Comment="HI"

:MemoryDisc,Comment,Group

Tagname1,,$System

Tagname2,,$System

因此，为确保有一个空字符串，必须在要保留为空的单元格中输入一个空格 （使用空格键）。下列关键字字段会受到如此影响：

Comment	**Eng Units**
OffMsg	**Initial Message**
OnMsg	**Application**
ItemName	**Topic**

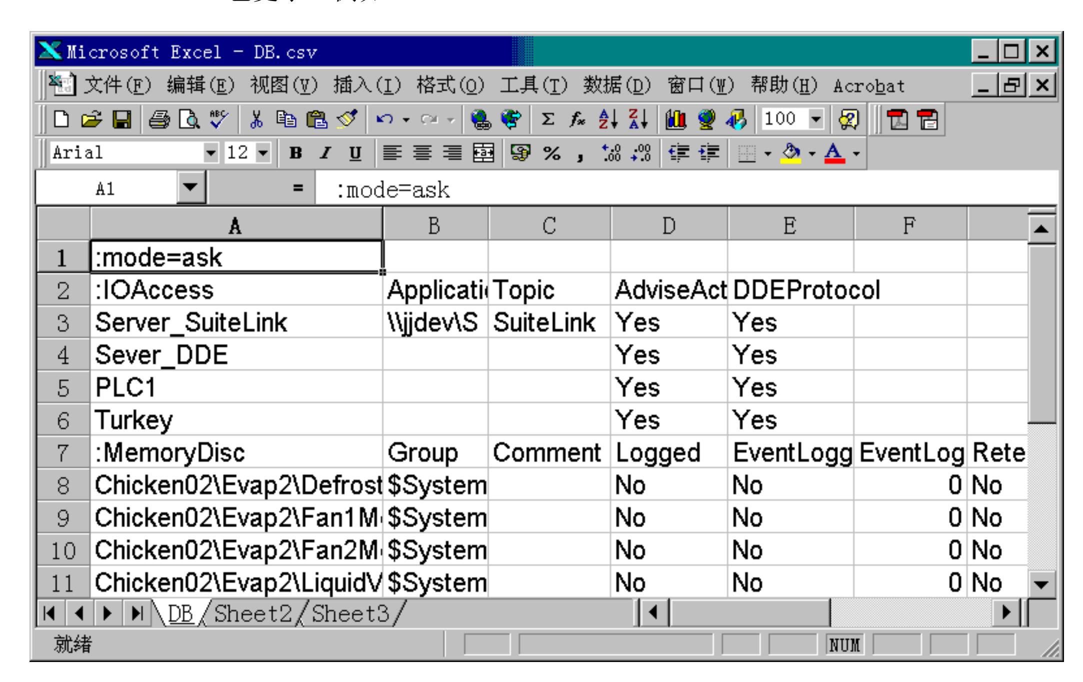
下面显示的是使用 Windows “记事本”创建的一个输入文件：

DB.CSV - 记事本

文件(F) 编辑(E) 格式(O) 帮助(H)

```
:mode=ask
:IOAccess,Application,Topic,AdviseActive,DDEProtocol
"jjdev1","\\jjdev\InTouch","test",Yes,Yes
"Server_SuiteLink","\\jjdev\SuiteLink","SuiteLink",Yes,Yes
"Server_DDE","\\jjdev\DDE","DDE",Yes,Yes
"PLC1","\\AN1\InTouch","PLC1",Yes,Yes
"Turkey","InTouch","Turkey",Yes,Yes
:MemoryDisc,Group,Comment,Logged,EventLogged,EventLoggingPriority,RetentiveValue,Initi
"Chicken02\Evap2\DefrostVlv","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
"Chicken02\Evap2\Fan1Motor","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
"Chicken02\Evap2\Fan2Motor","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
"Chicken02\Evap2\LiquidVlv","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
"Beef\EvapUnit2\DefrostVlv","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
"Beef\EvapUnit2\Fan1Motor","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
"Beef\EvapUnit2\Fan2Motor","$System","",No,No,0,No,Off,"","",None,1,"",0,0,""
```

使用某个程序（例如 Excel）创建输入文件时，除了在不同的列输入每一项之外，其余格式要求相同。这样读取数据会更加容易，出错的机会也更小。例如：



	A	B	C	D	E	F	
1	:mode=ask						
2	:IOAccess	Applicati	Topic	AdviseAct	DDEProtocol		
3	Server_SuiteLink	\\jjdev\S	SuiteLink	Yes	Yes		
4	Sever_DDE			Yes	Yes		
5	PLC1			Yes	Yes		
6	Turkey			Yes	Yes		
7	:MemoryDisc	Group	Comment	Logged	EventLogg	EventLog	Rete
8	Chicken02\Evap2\Defrost	$System		No	No	0	No
9	Chicken02\Evap2\Fan1M	$System		No	No	0	No
10	Chicken02\Evap2\Fan2M	$System		No	No	0	No
11	Chicken02\Evap2\LiquidV	$System		No	No	0	No

数据库输入文件操作 模式

下面列出有效的操作模式关键字，以及在加载期间遇到重复标记名时每种模式会采取的操作：

```
:MODE=REPLACE
:MODE=UPDATE
:MODE=ASK
:MODE=IGNORE
:MODE=TERMINATE
:MODE=TEST
```

:MODE=REPLACE

删除现有条目，代之以新条目。

:MODE=UPDATE

只使用输入文件中明确定义的字段改写现有的定义。

如果字段位于记录中，并且是由您输入或通过 "**:KEYWORD=value**" 机制进行设置的，则这些字段会被视作已明确定义。如果某个字段未在记录中指定，并且已经通过 "**:KEYWORD=**" 命令重置关键字，则当前字段值将不会更新。

下例描述在更新模式下的输入文件通过 DBLoad 加载 / 合并到应用程序的数据库时会发生的情况：

```
:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
; Tagname1 的组仅更新为 Group2
Tagname2,,
; Tagname2 的组更新为 Group1，而 Dconversion 保持不变
Tagname3,,Reverse
; Tagname3 的组更新为 Group1，而 Dconversion“取反”
; 下行将 Group 字段“重置”为它的缺省值
:Group=
;  数据字段 "Group" 重置为它的缺省值
Tagname4,,
; 只剩下 Tagname4
```

在 .csv 文件中允许使用注释。它们必须使用一个分号 (;) 加以注明。

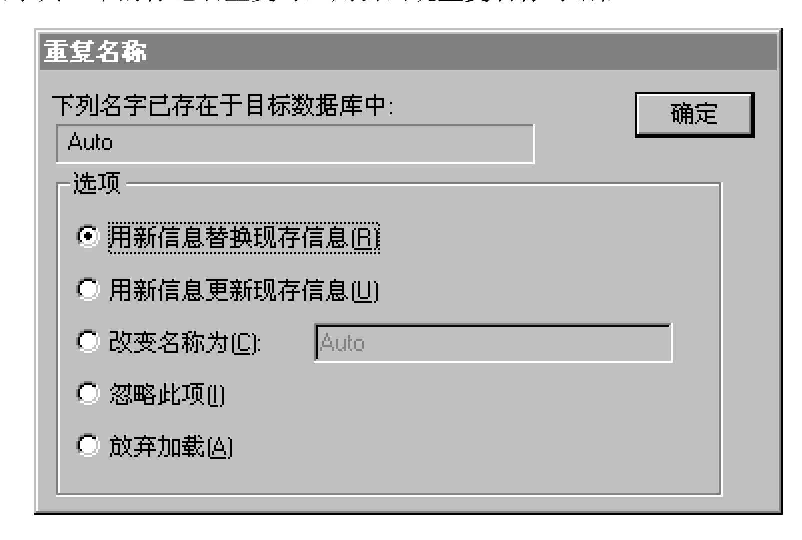
备注 如果要更改标记名类型并且该标记名正在使用中，则标记名类型必须兼容。例如，如果应用程序正在使用一个现有的历史趋势标记名，则不能将它改为 I/O 整型。此外，如果某个标记名正用在应用程序的输入链接中，则不能将它改为 **ReadOnly=yes**。由于这些限制，在运行 DBLoad 之前，您必须更新目标应用程序的标记名使用计数。

:MODE=ASK

将输入的或现有条目的标记名改变为指定的标记名，然后将新定义插入“标记名字典”。

备注 如果没有指定模式，则这是缺省输入模式。每次 DBLoad 遇到重复的标记名时，都会出现一个对话框，询问是否要替换现有的标记名。

操作模式设置为 **ASK** 并且输入文件中的标记名与目标应用程序 “标记名字典”中的标记名重复时，则会出现**重复名称**对话框：



选择某个选项，然后单击**确定**。加载完成时，会出现一个确认消息框。

选项	描述
改变名称为	使用在框中输入的名称替换指定标记名的名称。
忽略此项	忽略显示的标记名，处理过程继续进行。
用新信息替换现有信息	用新记录替换现有的标记名记录。
用新信息更新现有信息	只使用输入文件中明确定义的字段来改写现有的标记名记录。
放弃加载	取消加载功能。

备注 如果因某个问题导致加载失败，则会出现一个消息框。（错误消息将写入 Wonderware Logger 程序）。

:MODE=IGNORE

忽略新记录并继续处理。

:MODE=TERMINATE

终止处理。不更新目标文件。

:MODE=TEST

在这种模式下， DBLoad 如同在替换模式下运行，但不会修改数据库。它会在 Wonderware Logger 程序中报告发现的错误，并继续加载。这对于在实际处理之前验证输入文件的语法非常有用。

备注 将输入文件设置为测试模式之后，系统将忽略所有其它模式。您可以输入 **:mode=test** 作为文件的第一行，而不必去考虑文件中其它模式的改变。

如需有关操作模式的详细信息，请参阅 “创建数据库输入文件”。

创建数据库记录模板

关键字可用于创建模板记录，以便为后续数据记录中的各个数据字段提供 “全局”输入项。（除 **TAGNAME** 和 **TYPE** 字段以外，可以给每个字段值设置关键字）。

备注 模板记录必须遵照先前所述的有关创建逗号分隔输入文件的格式要求。例如，文件必须以 **:Type** 关键字条目开始。

设置字段值缺省值

关键字也可用于设置特定记录字段的缺省值。例如：

```
:KEYWORD=value
```

这会为所有后续的数据记录设置引用字段缺省值。此项功能可用于为许多记录中保持不变的字段值设置缺省值。如果某个字段定义了缺省值，则在记录中没有该值的数据时会使用缺省值。例如，**:GROUP=Reator_Site** 的结果是将所有具有空白 **GROUP** 项的标记名指定给 **Reator_Site** “报警组”。如果该标记名 （举例来说）给 **GROUP** 输入 **$System**，则它仍将指定给 “报警组” **$System**。

重置字段值缺省值

如果不指定任何值，则各个关键字可重置为它们原来的缺省值。例如，**:GROUP=**。

重置所有字段值缺省值

要重置所有关键字，请使用 **:RESET** 命令。此命令不带参数 (**:RESET**)，只要照原样输入就可以了，它会影响自己后面的所有项目。

备注 缺省值就是该标记名类型的原始 InTouch 值。例如，内存离散值使用 **Group=$System**、**EventLogging=Off**、**InitialValue=Off** 等作为其缺省值。

:Type 和关键字 条目

下表列出了每个标记名 **:Type** 的有效关键字，以及每个该类字段值的关键字。

:Type 和关键字	容许值	缺省值
:MemoryDisc		
ACKModel	整数 [0 = 条件， 1 = 事件或 2= 扩展]	0
AlarmComment	任意可读 / 写文本字符串 [适用于所有的标记类型。]	空白 **备注**：在 InTouch 7.1 或之前版本中，与标记名描述注释相同。
AlarmPri	1 到 999	1
AlarmState	无、开或关	无
Comment	任意文本字符串	""
DscAlarmDisable	1, 0	0
DscAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialDisc	1、 0、开、关、真、假、是或否	0
Logged	开、关、是或否	否
OffMsg	任意文本字符串	无
OnMsg	任意文本字符串	无
RetentiveValue	1、 0、开、关、真、假、是或否	否

:Type 和关键字	容许值	缺省值
:IODisc		
ACKModel	整数 [0 = 条件， 1 = 事件或 2= 扩展]	0
AlarmComment	任意可读 / 写文本字符串 [适用于所有的标记类型。]	空白 **备注**：在 InTouch 7.1 或之前版本中，与标记名描述注释相同。
AlarmPri	1 到 999	1
AlarmState	无、开或关	无
Comment	任意文本字符串	无
DConversion	直接或取反	直接
AccessName	有效访问名	无
DscAlarmDisable	1, 0	0
DscAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
组	有效的组名	$System
InitialDisc	1、 0、开、关、真、假、是或否	0
ItemName	有效的项目名	无
ItemUseTagname	真、假、是或否	是
Logged	开、关、是或否	否
OffMsg	任意文本字符串	无
OnMsg	任意文本字符串	无
ReadOnly	是或否	否
RetentiveValue	1、 0、开、关、真、假、是或否	否

:Type 和关键字	容许值	缺省值
:MemoryInt & :Memory Real		
ACKModel	整数 [0 = 条件，1 = 事件或 2= 扩展]	0
AlarmComment	任意可读 / 写文本字符串 [适用于所有的标记类型。]	空白 **备注：** 在 InTouch 7.1 或之前版本中，与标记名描述注释相同。
AlarmDevDeadband	有效偏差百分比	0
AlarmValueDeadband	有效的整型或实型值	0
Comment	任意文本字符串	无
Deadband	有效的整型或实型值	0
DevTarget	有效的整型或实型值	0
EngUnits	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
HiAlarmPri	1 到 999	1
HiAlarmState	是、否、开或关	否
HiAlarmValue	有效的整型或实型值	0
HiHiAlarmPri	1 到 999	1
HiHiAlarmState	是、否、开或关	否
HiHiAlarmValue	有效的整型或实型值	0
InitialValue	有效的整型或实型值	0
LoAlarmPri	1 到 999	1
LoAlarmState	是、否、开或关	否
LoAlarmValue	有效的整型或实型值	0
LogDeadband	有效的整型或实型值	0
Logged	开、关、是或否	否
LoLoAlarmPri	1 到 999	1
LoLoAlarmState	是、否、开或关	否
LoLoAlarmValue	有效的整型或实型值	0
MajorAlarmPri	1 到 999	1
MajorAlarmState	是、否、开或关	否
MajorAlarmValue	有效的整型或实型值	0
MaxValue	有效的整型或实型值	9999

:Type 和关键字	容许值	缺省值
MinorAlarmPri	1 到 999	1
MinorAlarmState	是、否、开或关	否
MinorDevAlarmValue	有效的整型或实型值	0
MinValue	有效的整型或实型值	0
RetentiveAlarmParameters	开、关、是或否	否
RetentiveValue	开、关、是或否	否
ROCAlarmPri	1 到 999	1
ROCAlarmState	是、否、开或关	否
ROCAlarmValue	任何有效的百分比	0
ROCTimeBase	秒、分或时	分

报警类型的报警约束标记名	容许值	缺省值
HiAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
HiHiAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
LoAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
LoLoAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
MajDevAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
MinDevAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
ROCAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无

停用标帜	容许值	缺省值
HiAlarmDisable	1, 0	0
HiHiAlarmDisable	1, 0	0
LoAlarmDisable	1, 0	0
LoLoAlarmDisable	1, 0	0
MajDevAlarmDisable	1, 0	0
MinDevAlarmDisable	1, 0	0
ROCAlarmDisable	1, 0	0

:Type 和关键字	容许值	缺省值
:IOInt & :IOReal		
ACKModel	整数 [0 = 条件， 1 = 事件或 2= 扩展]	0
AlarmComment	任意可读 / 写文本字符串 [适用于所有的标记类型。]	空白 **备注：**在 InTouch 7.1 或之前版本中，与标记名描述注释相同。
AlarmDevDeadband	有效偏差百分比	0
AlarmValueDeadband	有效的整型或实型值	0
Comment	任意文本字符串	无
Conversion	线性或平方根	线性
AccessName	有效访问名	无
Deadband	有效的整型或实型值	0
DevTarget	有效的整型或实型值	0
EngUnits	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
HiAlarmPri	1 到 999	1
HiAlarmState	是、否、开或关	否
HiAlarmValue	有效的整型或实型值	0
HiHiAlarmPri	1 到 999	1
HiHiAlarmState	是、否、开或关	否
HiHiAlarmValue	有效的整型或实型值	0
InitialValue	有效的整型或实型值	0
ItemName	有效的项目名	无
ItemUseTagname	真、假、是或否	否

:Type 和关键字	容许值	缺省值
LoAlarmPri	1 到 999	1
LoAlarmState	是、否、开或关	否
LoAlarmValue	有效的整型或实型值	0
LogDeadband	有效的整型或实型值	0
Logged	开、关、是或否	否
LoLoAlarmPri	1 到 999	1
LoLoAlarmState	是、否、开或关	否
LoLoAlarmValue	是、否、开或关	0
MajorDevAlarmPri	1 到 999	1
MajorDevAlarmState	是、否、开或关	否
MajorDevAlarmValue	有效的整型或实型值	0
MaxEU	有效的整型或实型值	9999
MaxRaw	有效的整型或实型值	9999
MinEU	有效的整型或实型值	0
MinorDevAlarmPri	1 到 999	1
MinorDevAlarmState	是、否、开或关	否
MinorDevAlarmValue	有效的整型或实型值	0
MinRaw	有效的整型或实型值	0
ReadOnly	是或否	否
RetentiveAlarmParameters	开、关、是或否	否
RetentiveValue	开、关、是或否	否
ROCAlarmPri	1 到 999	1
ROCAlarmState	是、否、开或关	否
ROCAlarmValue	任何有效的百分比	0
ROCTimeBase	秒、分或时	分

报警类型的报警约束标记名	容许值	缺省值
HiAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
HiHiAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
LoAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
LoLoAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
MajDevAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无

报警类型的报警约束标记名	容许值	缺省值
MinDevAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
ROCAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无

停用标帜	容许值	缺省值
HiAlarmDisable	1, 0	0
HiHiAlarmDisable	1, 0	0
LoAlarmDisable	1, 0	0
LoLoAlarmDisable	1, 0	0
MajDevAlarmDisable	1, 0	0
MinDevAlarmDisable	1, 0	0
ROCAlarmDisable	1, 0	0

:Type 和关键字	容许值	缺省值
:MemoryMsg		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialMessage	任意文本字符串	无
Logged	开、关、是或否	否
MaxLength	1-131 字符	131
RetentiveValue	开、关、是或否	否
Comment	任意文本字符串	无
AccessName	有效访问名	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
InitialMessage	任意文本字符串	无
ItemName	有效的项目名	无
ItemUseTagname	真、假、是或否	否
Logged	开、关、是或否	否
MaxLength	1-131 个字符	131
ReadOnly	是或否	否
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省值
:GroupVar		
Comment	任意文本字符串	无
Group	有效的组名	$System
EventLogged	开、关、是或否	否

:Type 和关键字	容许值	缺省值
:HistoryTrend		
Comment	任意文本字符串	无
Group	有效的组名	$System

:Type 和关键字	容许值	缺省值
:TagID		
Comment	任意文本字符串	无
Group	有效的组名	$System

:Type 和关键字	容许值	缺省值
:IndirectDisc		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省值
:IndirectAnalog		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省值
:IndirectMsg		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省值
:Access		
AdviseActive	是或否	是
NodeName	有效的网络节点名	无
Application	有效的 I/O 应用程序名	无
Topic	有效 I/O 主题名	无
DDEProtocol	“是”指定 DDE 协议； “否”指定 SuiteLink 协议	是

:Type 和关键字	容许值	缺省值
:AlarmGroup		
Comment	任意文本字符串	无
EventLogged	开、关、是或否	否
EventLoggingPriority	1 到 999	999
Group	有效的组名	$System
RetentiveValue	开、关、是或否	否

:Type 和关键字	容许值	缺省值
:AlarmDotfields		
AlarmAckModel	0, 1, 2	0
DscAlarmDisable	1, 0	0
DscAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
HiAlarmDisable	1, 0	0
HiHiAlarmDisable	1, 0	0
LoAlarmDisable	1, 0	0
LoLoAlarmDisable	1, 0	0
MajDevAlarmDisable	1, 0	0
MinDevAlarmDisable	1, 0	0
ROCAlarmDisable	1, 0	0
HiAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
HiHiAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
LoAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
LoLoAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
MajDevAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
MinDevAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无
ROCAlarmInhibitor	标记名引用：任意 “离散” 或 “模拟” 型标记名	无

第 7 章

创建动画链接

在创建图形对象或符号之后，您就可以通过动画来赋予其 “生命”。通过附加动画链接，您可以改变对象或符号的外观，以反映标记名或表达式值所发生的变化。例如，您可以创建一个泵符号，在它关闭时显示为红色，在它打开时显示为绿色。您也可以将此泵符号做成一个触动按钮，可供操作员通过单击鼠标或触动 （使用触摸屏时）来打开或关闭此泵。通过为对象或符号定义动画链接，您就能使用这些以及其它许多的特殊效果。

InTouch 支持两种基本类型的链接：“触动链接”和 “显示链接”。“触动链接”可供操作员向系统执行输入。“显示链接”则是给操作员显示输出结果。数值游标或按钮便是 “触动链接”的例子。而填充颜色、位置或闪烁链接则是 “显示链接”的例子。

本章介绍创建各类动画链接的操作步骤。

目录

通用动画链接功能

许多动画链接共享下面的通用控件：

- 对象类型对话框
- 通用调色板
- 快速访问 “标记浏览器”
- 快速访问 “标记名点域”
- 支持等价键

- 在**标记名**或**表达式**输入框中右击鼠标（显示一个菜单，包含可用于所选文本的各项命令）

对象类型对话框

对象类型对话框出现在屏幕顶部，在链接选择对话框之上。这是创建的所有链接都有的标题对话框。它显示您为动画链接附件所选对象类型的描述。例如**按钮**。

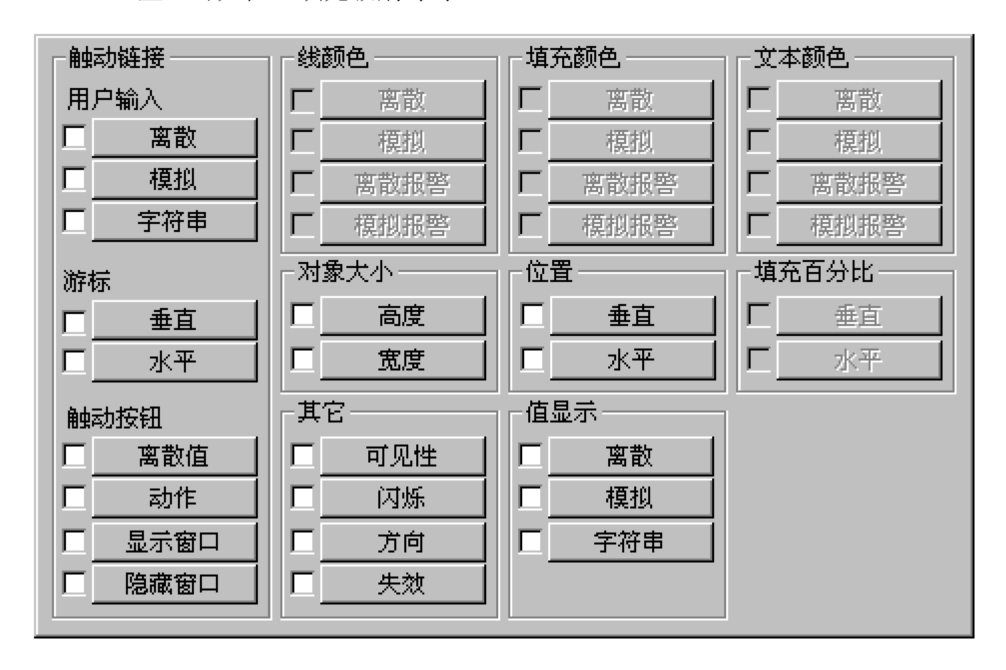


如果将多个链接附加到一个对象，则您可以单击**前一链接**和**后一链接**来快速向前或向后翻页，浏览对象所附加的每个链接所对应的链接对话框。

提示 链接是按照最初附加到对象的顺序存储的。

动画链接选择对话框

您可以为对象或符号定义多个链接。通过组合各种链接，您可以创建能想象得到的几乎任何屏幕动画效果。您可以改变对象的颜色、大小、位置、可见性、填充级别等等。



指定等价键

您可以指定用键盘上某个特定的键来激活一些动画链接。仅当包含此链接的对象可见或被选定时，等价键才可操作。

如果该对象包含可见性或禁用链接，则在该对象不可见或被禁用时，等价键将不会被激活。

您可以在多个窗口中定义相同的键。不过，只有最近打开的窗口中定义的键才会是活动的。在窗口重叠的情况下，顶部窗口的键才是活动的。

备注 如果将活动窗口中的任何对象或动作按钮指定给**键动作脚本**使用的同一个键，则活动窗口中该键的等价键链接的执行优先级高于**键动作脚本**。

如需有关“键动作脚本”的详细信息，请参阅第 8 章“在 InTouch 中创建 QuickScript”。

支持等价键的动画链接会在它们的链接对话框中显示**等价键**组。例如：



备注 直接选择键链接只会显示功能键 1-16。如果使用包含 16 个以上功能键的自定义键盘，则您需要从厂商处获取设备驱动程序，以允许访问系统上的扩展功能键。

要将键指定给链接

1. 如果希望操作员在按下等价键的同时按住 **Ctrl** 与 **Shift** 键中之一或两者，请选择 **Ctrl** 和（或）**Shift** 键。

2. 单击**键**。此时会出现**选择键**对话框。

选择键... 查找: 无

BackSpace	Up	Numpad1	Separator	F8	a	l	w	7
Tab	Right	Numpad2	Subtract	F9	b	m	x	8
Clear	Down	Numpad3	Decimal	F10	c	n	y	9
Return	Select	Numpad4	Divide	F11	d	o	z	无
Escape	Print	Numpad5	F1	F12	e	p	0	
Space	Execute	Numpad6	F2	F13	f	q	1	
PageUp	Copy	Numpad7	F3	F14	g	r	2	
PageDown	Insert	Numpad8	F4	F15	h	s	3	
End	Delete	Numpad9	F5	F16	i	t	4	
Home	Help	Multiply	F6	NumLock	j	u	5	
Left	Numpad0	Add	F7	CtrlBreak	k	v	6	

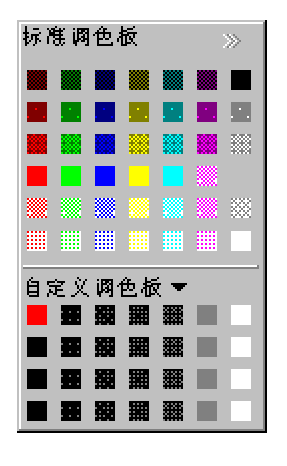
取消(C)

3. 单击要指定给该链接的键。此时该对话框会关闭，链接对话框再次出现，并在**键**按钮旁边显示所选键的名称。

应用颜色链接

您可以给线条、矩形、圆角矩形、椭圆、多边线、多边形以及文本的动态属性应用颜色。在给线条、填充或文本对象创建颜色链接时，您需要使用调色板来选择希望链接到标记名的值、标记名的报警状态等对象的颜色。

线颜色和文本颜色链接必须使用纯色。如果选择抖动 （混合）色，则缺省条件下， WindowMaker 会选择与之最接近的纯色。您可以创建自定义调色板，并将它们加载到标准的 WindowMaker 调色板中。



将颜色链接附加到对象或符号，并单击链接对话框中的颜色框时，则会出现调色板。

单击要用于该链接的颜色。此时调色板会自动关闭，所选颜色将显示在链接详细资料对话框的颜色框中。

如需有关自定义调色板的详细信息，请参阅第 1 章 “WindowMaker 程序元素”。

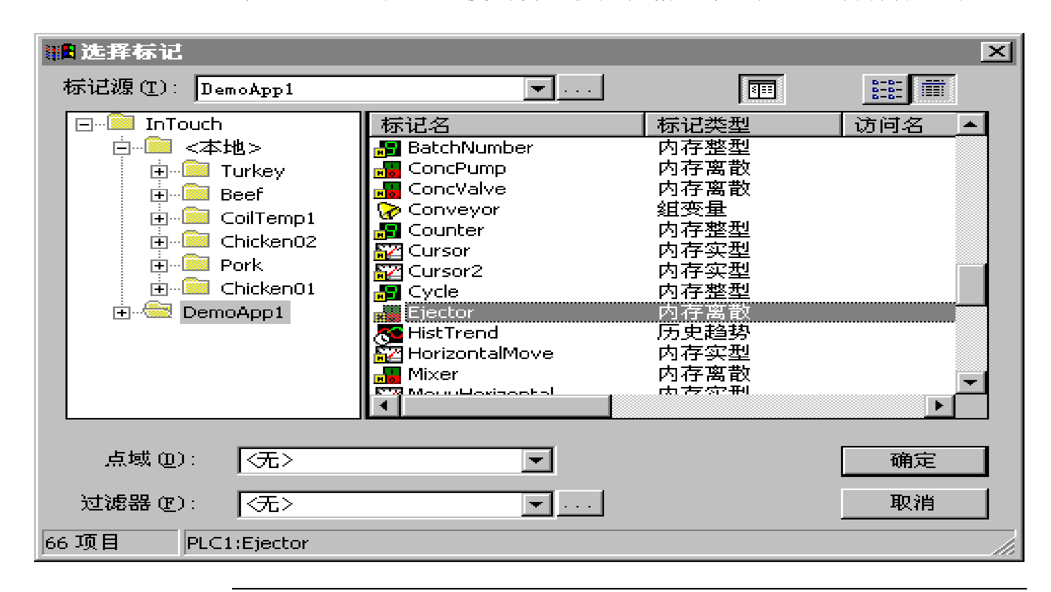
访问标记浏览器

通过访问 “标记浏览器”来创建动画链接时，您可以快速查看应用程序中定义的所有标记名。如果从 “标记浏览器”选择要指定给链接的标记名，它会自动插入**标记名**或**表达式**框。

要访问标记浏览器

1. 双击任意空白动画链接的**标记名**或**表达式**输入框。此时会出现 “标记浏览器”。

2. 单击 工具，以便用树形视图窗格显示已定义的所有标记源：



提示 如果未使用树形视图模式，请单击**标记源**箭头，选择要在列表中显示的标记源的名称。此时 “标记浏览器”将会刷新，并显示所选标记源的标记名。

3. 选择要给该链接使用的标记名，然后单击**确定**，或者双击标记名，以同时选择它、关闭 “标记浏览器”并将它插入**标记名**或**表达式**框。

提示 要给所选标记名使用**点域**，请单击**点域**箭头，从列表中选择要使用的**点域**，然后单击**确定**。

要显示某个标记名的数据库定义，请在**标记名**或**表达式**框中输入该标记名，然后双击它。此时会出现**标记名字典**对话框，显示该标记名的定义。

如需有关 “标记浏览器”的详细信息，请参阅第 6 章 “标记名字典”。

访问标记名点域

您可以通过两种方法从动画链接**标记名或表达式**输入框访问标记名**点域**。这两种方法现介绍如下。

要通过 “标记浏览器” 访问标记名 “点域”。

1. 双击空**标记名或表达式**输入框。此时会出现 “标记浏览器”，显示给当前标记源定义的各个标记名：



2. 单击**点域**箭头，以打开**点域**列表，从中您可以选择与当前所选标记名类型关联的点域。缺省条件下，所有类型的标记名最初均显示为 < 无 >。

3. 在列表中单击要追加到所选标记名的**点域**。

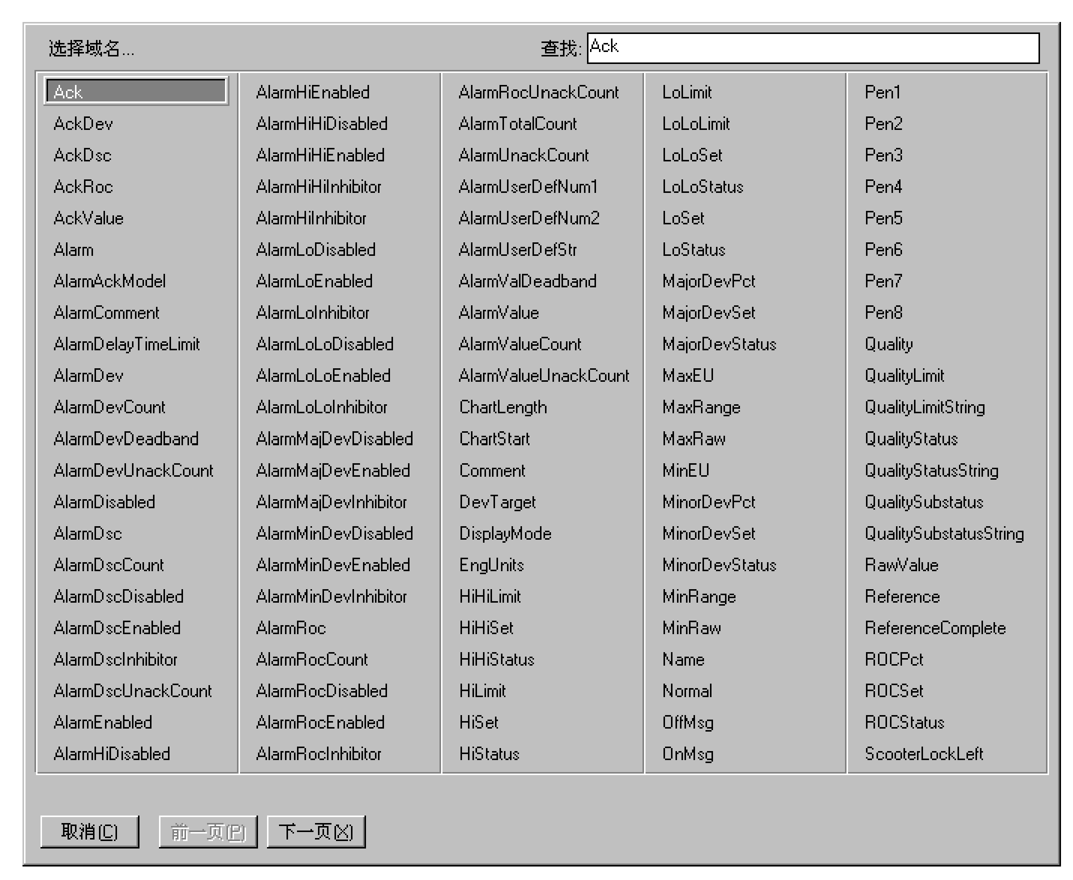
备注 并非每个标记名类型均有相同的**点域**。例如，**离散**型标记名有 **.OnMessage** 点域，而模拟型则没有。如果选择了一个**离散**型标记名，并给它指定了 **.OnMessage**，然后选择另一个**离散**型标记名，则显示的**点域**列表不会改变。但是，如果在控制视图列表中选择另一类型的标记名，例如模拟型，则显示的**点域**将恢复为 < 无 >。

如需有关 “标记浏览器” 的详细信息，请参阅第 6 章 “标记名字典”。

如需有关标记名**点域**的详细信息，请参阅联机 *Intouch 参考指南*。

要通过选择域名对话框来访问标记点域

1. 在任何**标记名**或**表达式**输入框中，输入标记名并加上一个英文句点（**tagname.**），然后双击它右侧；或者只输入一个英文句点，然后双击它右侧。此时会出现**选择域名**对话框，显示出所有的标记名点域：



2. 选择要使用的**点域**。此时该对话框会关闭，所选**点域**将自动插入**标记名**或**表达式**输入框。

动画对象

要设置对象或符号的动画效果

1. 创建并选择要设置动画的对象（线条、填充图案、文本、按钮或符号）。

2. 在**特别**菜单上，单击**动画链接**，或者双击该对象。此时会出现包含所有动画链接的对话框。

提示 您也可以右击对象，然后单击**动画链接**。

触动链接
用户输入：离散、模拟、字符串
游标：垂直、水平
触动按钮：离散值、动作、显示窗口、隐藏窗口

线颜色：离散、模拟、离散报警、模拟报警
填充颜色：离散、模拟、离散报警、模拟报警
文本颜色：离散、模拟、离散报警、模拟报警

对象大小：高度、宽度
位置：垂直、水平
填充百分比：垂直、水平

其它：可见性、闪烁、方向、失效
值显示：离散、模拟、字符串

3. 单击要将其附加到所选对象的链接的按钮。

提示 如果链接不适合所选的对象，则它的按钮将不会激活。

提示 单击复选框只能选择链接。通过单击链接名按钮来选择该链接并打开它的详细定义对话框。在您单击链接名按钮并接受输入时，复选框将自动选定。不过，如果清除某个链接的复选框，则动画链接会从所选对象中被移除。

4. 输入链接定义的详细资料，然后单击**确定**。此时会再次出现**链接选择**对话框，您可以视需要给该对象创建其它链接。

5. 单击**确定**以接受该对象的所有链接并关闭对话框。

提示 创建动画链接时，在将链接指定给您在动画链接的标记名或表达式框中输入的标记名之前，您必须先在“标记名字典”中定义它。如果未定义它，则会出现一个消息框，询问是否要立即定义该标记名。如果单击**是**，则会出现“标记名字典”，供您定义该标记名。

创建触动链接

触动链接用在您希望它在运行时是可以 “触控”的对象或符号上。它们可供操作员将数据输入系统。例如，操作员可以打开或关闭阀门、输入新的报警设定点、运行复杂的逻辑脚本，或者使用文本字符串登录，等等。

触动链接在运行时易于识别，因为在触控对象上移动光标，或按 TAB 键从一个对象移动到另一对象时，触控对象上总是被一个 “框”包围着。如果触动链接对象或符号包含重叠的文本对象，则最顶部的文本对象将用于显示数据值。

通过单击、触动对象 （使用触摸屏时）、按下指定的等价键，或者在对象被 “框”起时按下 **Enter** 键，操作员均可激活触控按钮。

您可以创建九种触动链接：

触动链接	动作
用户输入	离散、模拟、字符串
游标	垂直、水平
触动按钮	离散值、动作、显示窗口、隐藏窗口

备注 如果这些链接 （三维按钮除外）使用的对象或符号包含文本字段，则当文本对象显示在 WindowViewer 中时，会应用当前给文本设置的所有属性 （对齐方式、样式、字体等）。使用文本字段输入值时，屏幕也会显示输出值，除非在相应的标记名详细资料对话框中启用了**仅输入**选项。

下面几节介绍如何创建各种触动链接。

创建用户输入触动链接

您可以使用**用户输入触动链接**来创建可供操作员在系统中进行输入的触控对象。例如，使用按钮改变离散状态、模拟值或者安全性登录。

有三种类型的**用户输入**触动链接：

链接	描述
离散	用于控制离散型标记名的值。激活此链接时，会出现一个对话框，提示操作员作出选择。
模拟	用于输入模拟型 （整型或实型）标记名的值。激活此链接时，会出现一个输入框，可供您从标准键盘或可选的屏幕键盘输入值。
字符串	用于创建可输入字符串消息的对象。

备注 在运行期间，激活链接时，会出现一个用于输入框，用于输入消息值；或出现一个可选的屏幕键盘。

要创建离散输入链接

1. 双击对象，或选择对象然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**用户输入**区，单击**离散**。此时会出现**输入 -> 离散标记名**对话框。



3. 在**标记名**框中，输入离散型标记名。

   提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 如果要给链接指定一个等价键，请单击**键**。

   如需有关指定键的详细信息，请参阅指定等价键

5. 激活输入链接时，在**用户消息**框中，键入希望出现在输入对话框中的消息。
6. 在**置位提示**和**复位提示**框中，输入希望在按钮上显示的消息，操作员可在输入对话框中单击这些按钮来打开或关闭离散值。
7. 在**打开消息**和**关闭消息**框中，输入希望打开或关闭对象时出现在与该对象的文本字段 （如果有）中的消息。
8. 如果不想让输入内容显示在与对象关联的文本字段中，请选择**仅输入**。（此选项仅适于有文本字段与之关联的对象）。

9. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

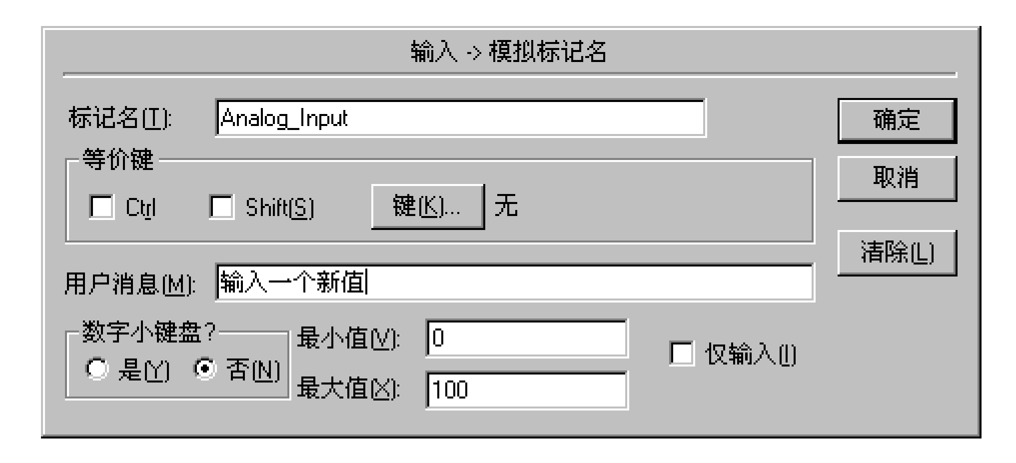
备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

要创建模拟输入链接

1. 双击对象，或选择对象然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**用户输入**区，单击**模拟**。此时会出现**输入 -> 模拟标记名**对话框。



   备注 如果此链接使用一个文本字段，则必须适当设置其格式以正确显示模拟 （整数或实数）值的输出。

   如需有关设置文本字段格式的详细信息，请参阅第 2 章 “使用 WindowMaker”。。

3. 在**标记名**框中，输入模拟型 （整型或实型）标记名。

   提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 要给链接指定等价键，请单击**键**。

   如需有关指定键的详细信息，请参阅 “指定等价键”。

5. 如果要在激活此链接时显示可选的数字小键盘，请在**用户消息**框中，输入希望在数字小键盘上出现的提示消息。

6. 如果希望在屏幕上显示一个数字小键盘来输入字符串的新值，请选择**是**。

7. 在**最小值**框中，给标记名输入最小输入值。
8. 在**最大值**框中，给标记名输入最大输入值。
9. 如果不想让输入内容显示在与对象关联的文本字段中，请选择**仅输入**。（此选项仅适于有文本字段与之关联的对象，例如，三维按钮）。
10. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 "标记名字典" 中定义 (远程标记名除外)，则系统会提示您立即去定义它。

要创建字符串输入链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**用户输入**区，单击**字符串**。此时会出现**输入 -> 字符串标记名**对话框。



3. 在**标记名**框中，输入一个消息型标记名。

   提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 如果要给链接指定一个等价键，请单击**键**。

   如需有关指定键的详细信息，请参阅 "指定等价键"。

5. 如果要在激活此链接时显示可选的数字小键盘，请在**用户消息**框中，输入希望在数字小键盘上显示的提示消息。

6. 如果希望输入的字符串照原样显示在屏幕上，请给**回显字符？**选项选择**是**选项。如果因数据保密 （例如，口令）而不应显示在屏幕上，请选择**否**。
7. 如果希望显示屏幕键盘以输入字符串的新值，请给**数字小键盘？**选项选择**是**选项。
8. 如果不想让输入内容显示在与对象关联的文本字段中，请选择**仅输入**。（此选项仅适于有文本字段与之关联的对象，例如，三维按钮）。

   备注 初次启动 WindowViewer 时，该字符串会显示定义与此对象链接的标记名时在**初始值**框中输入的文本。

9. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

创建游标触动链接

您可以使用**游标触动链接**来创建对象或符号，通过使用鼠标或其它定位设备 （例如，用手指在触模屏上），可以在窗口上移动这些对象或符号。在对象或符号移动时，与之链接的标记名的值会发生改变。这样您就可以创建用于在系统中设置值的设备。

一个对象可以有水平或垂直游标触动链接，或同时具有两种链接。通过在单个对象上使用两种链接，您可以同时改变两个模拟型标记名的值。

备注 水平和垂直游标链接是使用相同的方法创建的。此操作程序介绍**水平游标**链接。

要创建水平 （或垂直）游标链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**游标**区，单击**水平**。此时会出现**水平游标**对话框。



3. 在 “标记名”框中，输入模拟型 （整型或实型）标记名。

提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 在**在左端**框中，输入游标到达最左端位置时标记名的值。
5. 在**在右端**框中，输入游标到达最右端位置时标记名的值。
6. 在**向左**框中，输入游标可向左移动的象素数。

提示 在最左端，标记名的值等于**在左端**字段中输入的值。

7. 在**向右**框中，输入游标可向右移动的象素数。

提示 在最右端，标记名的值等于在**在右端**字段中输入的值。

8. 选择对象上的**引用位置**，移动对象时光标将锁定在该位置。
9. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

创建触动按钮触动链接

您可以使用**触动按钮触动链接**来创建对象链接，在使用鼠标单击它们，或是在触动 （使用触摸屏时）它们时，这些对象链接将立即执行某个操作。这些操作可以是**改变离散值**、执行**动作脚本**、**显示**窗口或**隐藏窗口**命令。**触动按钮**链接有四种类型：

链接	描述
离散值	用于将任何对象或符号设置成控制离散型标记名状态的按钮。这些按钮动作可以是设置、重置、切换、瞬间打开 （直接）和瞬间关闭 （取反）等类型。
动作	允许任何对象、符号或按钮链接最多三种不同的动作脚本：按下时、按下期间和释放时。动作脚本可用于将标记名设置为特定的值、显示和 （或）隐藏窗口、启动和控制其它应用程序、执行函数等。
显示窗口	用于将对象或符号设置成在被单击或触动时可打开一个或多个窗口的按钮。
隐藏窗口	用于将对象或符号设置成被单击或触动时可关闭一个或多个窗口的按钮。

要创建离散值触动按钮链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**触动按钮**区，单击**离散值**。此时会出现**按钮 -> 离散值**对话框。



3. 在**标记名**框中，输入离散型标记名。

   提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 如果要给链接指定一个等价键，请单击**键**。

5. 如需有关指定键的详细信息，请参阅 “给动画链接指定键”。
6. 选择要用于按钮的**动作**选项，具体如下：

动作	描述
直接	只要按下或按住按钮，便将值设置为 1（True、On、Yes）。释放按钮时，该值会自动复位成 0 （False、 Off、 No）。
取反	按下或按住按钮时，将值设置为 0 （False、 Off、 No）。释放按钮时，该值会自动复位成 1 （True、 On、 Yes）。
切换	按下按钮时，反转离散型标记名的状态。例如，如果标记名为 1 时按下按钮，则它会重置为 0，反之亦然。
重置	按下按钮时，将值设置为 0 （False、 Off、 No）。
置位	按下按钮时，将值设置为 1 （True、 On、 Yes）。

7. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您去立即定义它。

要创建动作触动按钮链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**触动按钮**区，单击**动作**。此时会出现 **InTouch -> 动作脚本**编辑器。



如需有关编写 QuickScript 的详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

3. 单击**条件类型**箭头，选择要应用于对象的脚本类型。您可以给相同的键应用所有这三种类型的脚本：

脚本	描述
按下时	初次按下键时，执行一次脚本。
按下期间	只要按住键，便按一定的时间间隔连续执行脚本。
放开时	释放键时，执行一次脚本。

提示 **按下期间**脚本在指定的毫秒数之后开始执行。要立即执行脚本，则需创建一个相同**按下时**脚本。

备注 如果活动窗口中的任何对象或动作按钮被指定给**键脚本**所用的同一个键，则此键在活动窗口中的等价键链接将优先于**键动作脚本**执行。

如需有关脚本的详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

4. 在脚本编辑器窗口中单击鼠标按钮，然后输入希望在激活对象时执行的脚本。
5. 单击**确定**将脚本附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

要创建显示 （或隐藏）窗口触动按钮链接

备注 **显示窗口**和**隐藏窗口**链接是使用相同的方法来创建的。此操作程序向您介绍**显示窗口**链接。

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**触动按钮**区，单击**显示窗口**。此时会出现**当触动时显示窗口** 对话框。



3. 选择在单击或触动对象时要打开的窗口。
4. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

提示 如果屏幕显示一个以上的窗口，而其中的一个窗口为**替换**型，则与之相交的任何其它窗口将会在它们显示之前关闭（给您造成**显示窗口**动画链接失效的错觉）。

要改变窗口类型，请右击打开的窗口的空白区，然后单击**窗口属性**。此时会出现**窗口属性**对话框，此时您便可以改变类型了。（如果 WindowViewer 正在运行，则不能改变窗口类型）。

如需有关窗口属性的详细信息，请参阅第 2 章 “使用 WindowMaker”。

创建显示链接

您可以使用各种**显示链接**向操作员提供输出。可供创建的显示链接有八种类型：

显示链接	类型
线颜色、填充颜色和文本颜色	离散、模拟、离散报警、模拟报警
对象大小	高度、宽度
位置	水平、垂直
填充百分比	水平、垂直
其它	可见性、方向、闪烁、失效
值显示	离散、模拟、字符串

下面几节介绍如何创建各种显示链接。

创建颜色链接

您可以使用颜色链接该对象的**线颜色**、**填充颜色**和**文本颜色**属性设置动画效果。

备注 线颜色和文本颜色链接必须使用纯色。如果选择抖动（混合）色，则缺省条件下， WindowMaker 会选择与之最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度设置必须高于 256 色，例如， 32K 或 65K （有时也叫 “增强色”）。

如需有关调色板的详细信息，请参阅 “应用颜色链接”。

上述每种颜色属性均可通过定义属性的颜色链接进行动态设置。颜色属性可以与离散表达式、模拟表达式、离散报警状态或模拟报警状态的值链接。

线颜色、填充颜色和文本颜色有四种类型：

颜色链接	描述
离散	用于控制与离散表达式的值链接的对象或符号的填充颜色、线颜色和文本颜色属性。
模拟	对象或符号的线颜色、填充颜色和文本颜色可以与模拟标记名（整型或实型）或模拟表达式的值链接。通过指定四个断点，可以定义五个值范围。您可以选择随值范围改变而显示的五种不同颜色。
离散报警	对象的文本、线和填充颜色均可与标记名、“报警组”或“组变量”的报警状态链接。此颜色链接可供选择两种颜色，一种在标记名处于标准状态下使用，另一种则在标记名处于报警状态下使用。此链接可同时用于模拟型和离散型这两种标记名。如果用于模拟型标记名，则它可以响应标记名的任何报警条件。
模拟报警	对象的文本、线和填充颜色均可与模拟型标记名、“报警组”或“组变量”的报警状态链接。可供给标准状态设置一种特定的颜色，同时也可供为标记名定义的每种报警条件设置一种不同的颜色。 **备注** 使用“模拟报警”动画链接，且该动画链接使用的远程标记名正在访问 InTouch 7.0 版以前版本所创建且未转换的应用程序包含的标记名信息，则对象不会进入报警状态。

要创建离散填充颜色链接

备注 所有的**线颜色**和**文本颜色**链接都是使用与**填充颜色**相同的方法创建的。下列操作程序介绍如何创建**填充颜色**链接。

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

    提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**填充颜色**（或线颜色或文本颜色）区，单击**离散**。此时会出现**填充颜色 -> 离散表达式**对话框。



3. 在**表达式**框中，输入等于真或假的离散型标记名或表达式。

提示 离散表达式也可以包含模拟型标记名。例如， TankLevel >= 75。在本例中，当变量 "TankLevel" 值大于或等于 "75" 时，对象的填充颜色将会改变。

右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 “触发器” 标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接” 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunctions 的详细信息，请参阅 第 8 章 “在 InTouch 中创建 QuickScript”。

4. 在**颜色**组中，单击每个颜色方框以打开调色板。单击调色板中要用于每种标记名状态的颜色。

备注 线颜色和文本颜色链接必须使用纯色。如果选择抖动 （混合）色，则缺省条件下， WindowMaker 会选择与之最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度设置必须高于 256 色，例如， 32K 或 65K （有时也叫 “增强色”）。

如需有关调色板的详细信息，请参阅 “应用颜色链接”。

5. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

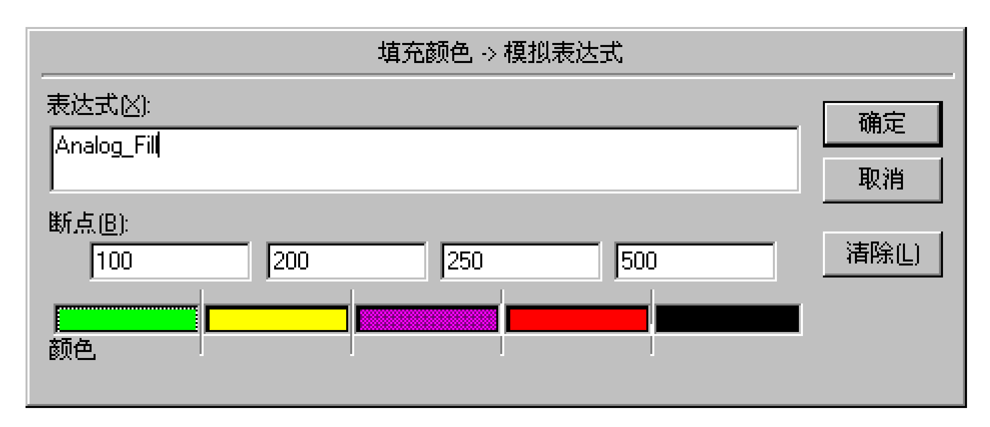
备注 如果输入的标记名未在 “标记名字典” 中定义 （远程标记名除外），则系统会提示您立即去定义它。

要创建模拟表达式颜色链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**填充颜色**（或**线颜色**或**文本颜色**）区，单击**模拟**。此时会出现**填充颜色 -> 模拟表达式**对话框。



3. 在**表达式**框中，输入模拟型（整型或实型）标记名或等于模拟值的表达式。

提示 右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 “触发器”标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接”的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunctions 的详细信息，请参阅 第 8 章 “在 InTouch 中创建 QuickScript”。

4. 在每个**断点**框中，您可以指定对象将在该处改变颜色的断点值（对于实型标记名而言，小数是有效的）。

提示 您不必使用 4 个不同的值。例如，如果只希望对象改变颜色 3 次，请输入 3 个值，而对第 3 和第 4 个值使用相同的颜色。

5. 在**颜色**组中，单击每个颜色方框以打开调色板。单击调色板中要用于每个断点的颜色。

备注 线颜色和文本颜色链接必须使用纯色。如果选择抖动（混合）色，则缺省条件下， WindowMaker 会选择与之最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度设置必须高于 256 色，例如， 32K 或 65K （有时也叫 “增强色”）。

如需有关调色板的详细信息，请参阅 “应用颜色链接”。

6. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

要创建离散报警状态颜色链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**填充颜色** （或**线颜色**或**文本颜色**）区，单击**离散报警**。此时会出现**填充颜色 -> 离散标记名报警状态**对话框。



3. 在**标记名**框中，输入要将其报警状态与对象关联的离散型标记名。

提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 在**颜色**组中，单击每个颜色方框以打开调色板。单击调色板中要用于每种颜色状态的颜色。

备注 线颜色和文本颜色链接必须使用纯色。如果选择抖动（混合）色，则缺省条件下， WindowMaker 会选择与之最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度设置必须高于 256 色，例如， 32K 或 65K （有时也叫 “增强色”）。

如需有关调色板的详细信息，请参阅 “应用颜色链接”。

5. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

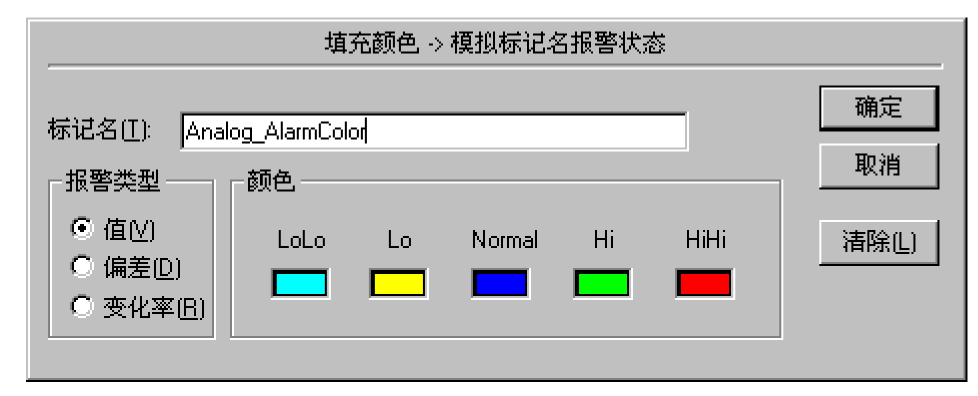
备注 如果输入的标记名未在“标记名字典”中定义（远程标记名除外），则系统会提示您立即去定义它。

要创建模拟报警状态颜色链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**填充颜色**（或**线颜色**或**文本颜色**）区，单击**模拟报警**。此时会出现**填充颜色 -> 模拟标记名报警状态**对话框。



3. 在**标记名**框中，输入要将其报警状态与对象关联的模拟型（整型或实型）标记名。

   提示 右击**标记名**框，以访问可应用于所选文本的命令。

4. 在**报警类型**组中，选择要与对象关联的报警类型。您可以使用三种互相独立的模拟颜色链接类型：

报警类型	描述
值报警	根据为标记名（见上例）定义的值报警的状态，您最多可以选择 5 种不同颜色。
偏差	根据为标记名（见上例）定义的偏差报警的状态，您最多可以选择 3 种不同颜色。
ROC（变化率）	**ROC**（Rate of Change，变化率）- 根据为标记名（见上例）定义的变化率报警的状态，您最多可以选择 2 种不同颜色。

5. 在**颜色**组中，单击每个颜色方框以打开调色板。单击调色板中要用于每种颜色状态的颜色。

备注 线颜色和文本颜色链接必须使用纯色。如果选择抖动（混合）色，则缺省条件下，WindowMaker 会选择与之最接近的纯色。为避免混合颜色，显卡的最低配置必须为 2MB，颜色深度设置必须高于 256 色，例如，32K 或 65K（有时也叫“增强色”）。

如需有关调色板的详细信息，请参阅“应用颜色链接”。

6. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

创建对象大小链接

您可以使用**对象大小**链接来根据模拟型（整型或实型）标记名或模拟表达式的值改变对象的高度和（或）宽度。“对象大小”链接通过设置链接的“参考点”，控制对象高度和（或）宽度及伸缩方向。高度和宽度链接可以附加到同一个对象。

备注 高度和宽度链接是使用相同方式创建的。此操作程序介绍**高度**链接。

要创建高度（或宽度）链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

---

2. 在**对象大小**区，单击**高度**。此时会出现**对象高度 -> 模拟量**对话框。



3. 在**表达式**框中，输入模拟型（整型或实型）标记名或等于模拟值的表达式。

提示 右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 "触发器" 标记名用作 QuickFunction 的参数，以强制更新 "动画链接"。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 "动画链接" 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅第 8 章 "在 InTouch 中创建 QuickScript"。

4. 在**最大高度值**框中，输入将使对象达到最大高度的标记名或表达式的值。
5. 在**最小高度值**框中，输入将使对象达到最小高度的标记名或表达式的值。
6. 在**最大 % 高度**框中，输入在标记名或表达式达到**最大高度值**字段中设置的值时，对象高度的百分比 (0-100)。
7. 在**最小 % 高度**框中，输入在标记名或表达式达到**最小高度值**字段中设置的值时，对象的高度百分比 (0-100)。

提示 高度百分比数字是以对象的实际 "绘制大小"（设置为 100%）的百分比来表示的。

8. 选择对象高度开始增长的**锚**点。

提示 选择**顶部**将导致对象从顶端往下扩大。选择**中间**将导致对象从中间点向两边扩展。选择**底部**将导致对象从底端往上扩展。

9. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 "标记名字典" 中定义 （远程标记名除外），则系统会提示您立即去定义它。

创建位置链接

您可以使用**位置链接**让对象自动水平、垂直或同时朝两个方向移动，以对应模拟型标记名或表达式的值所发生的变化。

备注 **水平位置**和**垂直位置**链接是以相同的方式创建的。此操作程序介绍**水平位置**链接。

要创建水平位置链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**位置**区，单击**水平**。此时会出现**水平位置**对话框。



3. 在**表达式**框中，输入模拟型 （整型或实型）标记名或等于模拟值的表达式。

   提示 右击**表达式**框，以访问可应用于所选文本的各个命令。

   备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

   您必须将一个 “触发器”标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接”的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

   如需有关 QuickFunction 的详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

4. 在**在左端**框中，输入对象位于最左端时的标记名的值。

5. 在**在右端**框中，输入对象位于最右边时的标记名的值。
6. 在**到左边**框中，输入对象可向绘制位置左侧移动的象素数。

提示 在最左端，标记名的值等于**在左端**字段中输入的值。

7. 在**到右边**框中，输入对象可向绘制位置右侧移动的象素数。

提示 在最右端，标记名的值等于在**在右端**字段中输入的值。

8. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

创建填充百分比链接

您可以使用**填充百分比链接**，以根据模拟型标记名的值或用于计算模拟值的表达式，来改变填充图案 （或包含填充图案的符号）的填充级别。例如，此链接可用于显示容器中的液面高度。对象或符号可以使用水平填充链接或垂直填充链接，或同时使用这两种链接。

备注 **水平填充百分比**和**垂直填充百分比**链接是使用相同的方式创建的。此操作程序介绍**垂直填充百分比**链接。

要创建垂直填充百分比链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**填充百分比**区，单击**垂直**。此时会出现**垂直填充 -> 模拟量**对话框。

3. 在**表达式**框中，输入模拟型（整型或实型）标记名或等于模拟值的表达式。

提示 右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个“触发器”标记名用作 QuickFunction 的参数，以强制更新“动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对“动画链接”的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅第 8 章“在 InTouch 中创建 QuickScript”。

4. 在**最大填充数的值**框中，输入将使对象填充到最大级别的表达式的值。

5. 在**最小填充数的值**框中，输入将使对象填充到最小级别的表达式的值。

6. 在**最大 % 填充**框中，输入在表达式达到**最大填充数的值**框中设置的级别时，对象填充的百分比 (0-100)。

提示 如果表达式的值大于这个数字，则它会被忽略。

7. 在**最小 % 填充**框中，输入在表达式达到**最小填充数的值**框中设置的级别时，对象填充的百分比 (0-100)。

提示 如果表达式的值大于这个数字，则它会被忽略。

8. 选择要填充对象的**方向**。

提示 如果选择**上**，则从下向上填充对象。如果选择**下**，则从上向下填充对象。

9. 在**背景颜色**框中，您可以打开调色板。单击所需的颜色。此时调色板会从屏幕上消失。

提示 选择的此种**背景颜色**会用作该对象**未填充**部分的颜色。实际填充的颜色是绘制对象时选择的颜色。如果同时将**垂直填充百分比**和**水平填充百分比**链接链接到相同的对象，则在其中任一链接对话框中最后选择的颜色将用作背景颜色。

10. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典”中定义 （远程标记名除外），则系统会提示您立即去定义它。

创建其它链接

系统还提供了四种其它类型的链接。

其它链接	描述
可见性	用于根据离散型标记名或表达式的值来控制对象的可见性。
闪烁	用于根据离散型标记名或表达式的值来使对象闪烁。
方向	用于根据标记名或表达式的值来旋转对象。
失效	用于根据标记名或表达式的值了禁用对象的触动功能。

提示 常用作安全策略的一部分。

如需有关在应用程序中应用安全设置的详细信息，请参阅第 5 章 “建立分布式应用程序”。

要创建可见性链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击该对象，然后再单击“动画链接”。

2. 在**其它**区，单击**可见性**。此时会出现**对象可见性 -> 离散值**对话框。



3. 在**表达式**框中，输入离散型标记名或等于离散值的表达式。

提示 离散表达式也可以包含模拟型标记名，例如 TankLevel >= 75。在本例中，当标记名 TankLevel 的值大小或等于 75 时，就可以在窗口中看见对象。

右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个“触发器”标记名用作 QuickFunction 的参数，以强制更新“动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对“动画链接”的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅 第 8 章“在 InTouch 中创建 QuickScript”。

4. 选择对象的**可见状态**。如果选择**打开**，则在表达式的值为真时，对象就不可见。如果选择**关闭**，则在表达式的值为真时，对象便是可见的。

5. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

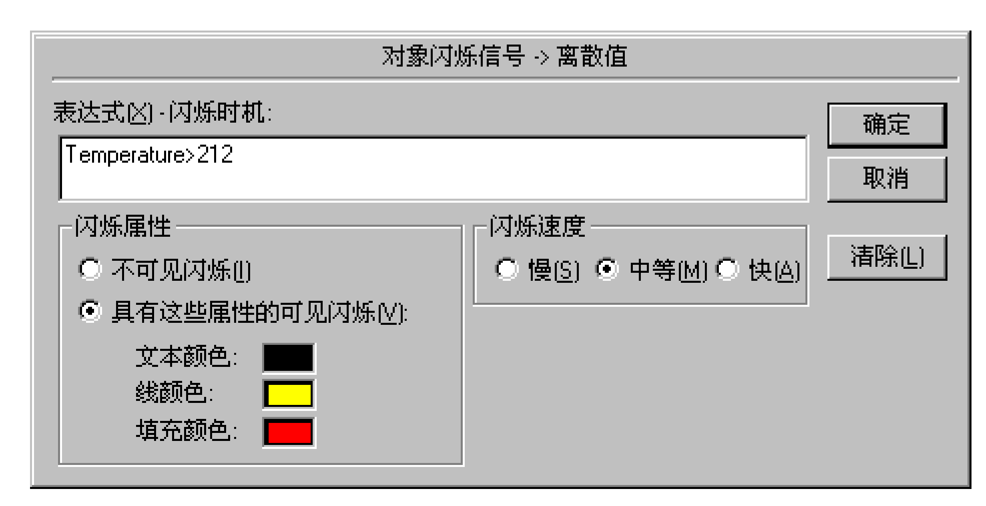
备注 如果输入的标记名未在“标记名字典”中定义（远程标记名除外），则系统会提示您立即去定义它。

要创建闪烁链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**其它**区，单击**闪烁**。此时会出现**对象闪烁信号 -> 离散值**对话框。



3. 在**表达式 - 闪烁时机**框中，输入离散型标记名或等于离散值的表达式。

   提示 离散表达式也可以包含模拟型标记名。例如 TankLevel >= 75。在本例中，在变量 "TankLevel" 的值大于或等于 "75" 时，该对象将会闪烁。

   右击**表达式**框，以访问可应用于所选文本的各个命令。

   备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

   您必须将一个 “触发器” 标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接” 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

   如需有关 QuickFunction 的详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

4. 为该对象选择所需的**闪烁属性**。

   如果选择**不可见闪烁**，则对象（或符号）的闪烁方式是在窗口中消失然后重新出现。如果选择**具有这些属性的可见闪烁**，则对象（或符号）将在窗口中保持可见，而改变所选颜色属性会产生闪烁效果。

   单击**文本颜色**、**线颜色**和**填充颜色**框以打开调色板。单击所需的颜色。此时调色板会从屏幕上消失。

   提示 选择与对象的"填充"颜色相同的"填充"闪烁颜色可以使对象不"闪烁"。

5. 给对象选择所需的**闪烁速度**。

   提示 要将闪烁速度配置为**慢**、**中等**和**快**，请在**特别**菜单上，指向**配置**，然后单击 **WindowViewer**。此时会出现 **WindowViewer 通用**属性页。在**闪烁频率**组中，输入要用于表示速度的毫秒数。

   对这些设置所作的任何更改都是全局性的，因此会影响整个应用程序中所有闪烁链接的闪烁速度。

6. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在"标记名字典"中定义（远程标记名除外），则系统会提示您去立即定义它。

要创建方向链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**其它**区，单击**方向**。此时会出现**方向 -> 模拟量**对话框。

方向 -> 模拟量

表达式(E):
Fan_Blade
确定(O)
取消
属性
最大逆时针旋转时的值(V): 1 逆时针旋转(C) 1
最大顺时针旋转时的值(M): 36 顺时针旋转(W) 360
清除(L)
旋转中心相对于对象中点的偏移量
X 轴坐标(X): 0 Y 轴坐标(Y): 0

3. 在**表达式**框中，输入模拟型 （整型或实型）标记名或等于模拟值的表达式。

提示 右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 “触发器” 标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接” 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅 第 8 章 “在 InTouch 中创建 QuickScript”。

4. 在**最大逆时针旋转时的值**框中，输入要使对象旋转到最大逆时针位置的表达式值。

提示 如果表达式的值大于这个数字，则它会被忽略。

5. 在**最大顺时针旋转时的值**框中，输入要使对象旋转到最大顺时针位置的表达式值。

提示 如果表达式的值大于这个数字，则它会被忽略。

6. 在**逆时针旋转**框中，输入到达**最大逆时针旋转时的值**时，对象逆时针旋转的度数。

7. 在**顺时针旋转**框中，输入到达**最大顺时针旋转时的值**时，对象顺时针旋转的度数。

提示 对象将根据在 WindowMaker 中绘制时的原始位置进行顺时针或逆时针旋转。

要强制对象（如文本）旋转特定的角度，只要将**最大逆时针旋转时的值**设置为 360、将**最大顺时针旋转时的值**设置为 0、将**逆时针旋转**设置为 360、将**顺时针旋转**设置为 0，然后在**表达式**框中输入角度值，如 90（对应 90 度角）。请注意，如果未定义标记名，此表达式永远也不会改变，而且对象总保持在 90 度位置。

文本可以在 WindowMaker 中设置，但不能在 WindowMaker 中按标记名的值进行旋转。

8. 在 **X 轴坐标**框中，输入旋转中心点从对象中心点水平移动的象素数。（正数值在中心点的右侧）。

提示 方向链接将对象或符号的中心用作旋转中心。

9. 在 **Y 轴坐标**框中，输入旋转中心点从对象中心点垂直移动的象素数。（正数值位于中心点的左侧）。
10. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在“标记名字典”中定义（远程标记名除外），则系统会提示您立即去定义它。

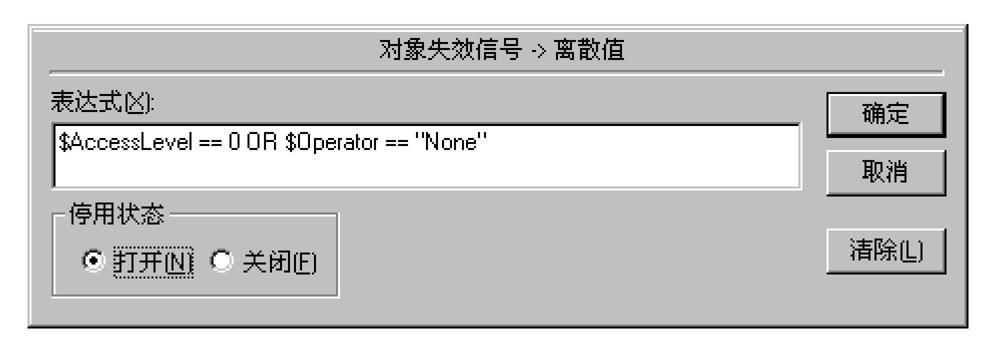
要创建失效链接

提示 如果要在应用程序中应用安全机制，则失效链接将很有用处。例如，您可以根据登录操作员的访问级别或用户名来使某些对象失效。

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**其它**区，单击**失效**。此时会出现**对象失效信号 -> 离散值**对话框。



3. 在**表达式**框中，输入离散型标记名或等于离散值的表达式。

提示 通过使用上面的表达式，如果无人登录，则对象或按钮会被禁用，以防被擅自篡改。

离散表达式也可以包含模拟型标记名。例如， TankLevel >= 75。在本例中，在变量 "TankLevel" 的值大于或等于 "75" 时，对象将失效。

右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 “触发器” 标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接” 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅第 8 章 “在 InTouch 中创建 QuickScript”。

4. 选择**失效状态**，在离散型标记名或表达式为真时，它将关闭或开启对象的功能。

提示 停用状态 “打开” 表示只要表达式为真，该对象或按钮的触动功能便会被关闭，因此无法单击它。

5. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典” 中定义 （远程标记名除外），则系统会提示您立即去定义它。

创建值显示链接

“值显示链接”提供了使用文本对象显示离散、模拟或字符串标记名的值的功能。共有三种类型：

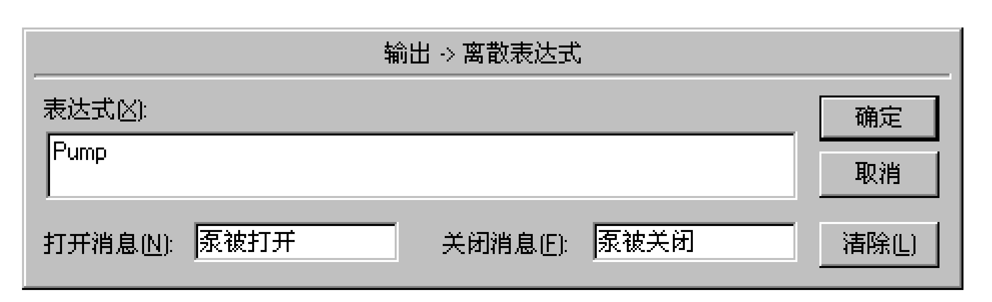
值显示类型	描述
离散	使用离散表达式的值在文本对象中显示 “打开”或 “关闭”用户自定义消息。
模拟	在文本对象中显示模拟表达式的值。
字符串	在文本对象中显示字符串表达式的值。

要创建离散值显示链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

   提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**值显示**区，单击**离散**。此时出现**输出量 -> 离散表达式**对话框。



3. 在**表达式**框中，输入离散型标记名或等于离散值的表达式。

   提示 离散表达式也可以包含模拟型标记名。例如， TankLevel >= 75。在本例中，当变量 "TankLevel" 的值大于或等于 "75" 时，将显示相应的消息。

   右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 “触发器” 标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接” 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅 第 8 章 “在 InTouch 中创建 QuickScript”。

4. 在**打开消息**框中，输入在离散表达式的值等于 1 （True、 On、 Yes） 时想要显示的消息。
5. 在**关闭消息**框中，输入在离散表达式的值等于 0 （False、 Off、 No） 时想要显示的消息。

提示 消息会显示在原始文本对象所在位置，使用为该对象设置的字体、大小、颜色、对齐方式及链接的属性。字段的原始内容不会影响运行时显示的消息。

6. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典” 中定义 （远程标记名除外），则系统会提示您立即去定义它。

提示 您也可以使用**值显示输出 -> 字符串表达式**链接来显示离散标记名的打开或关闭消息。对于此链接，您需要输入下面的表达式：

```
DText (Pump, Pump.OnMsg, Pump.OffMsg);
```

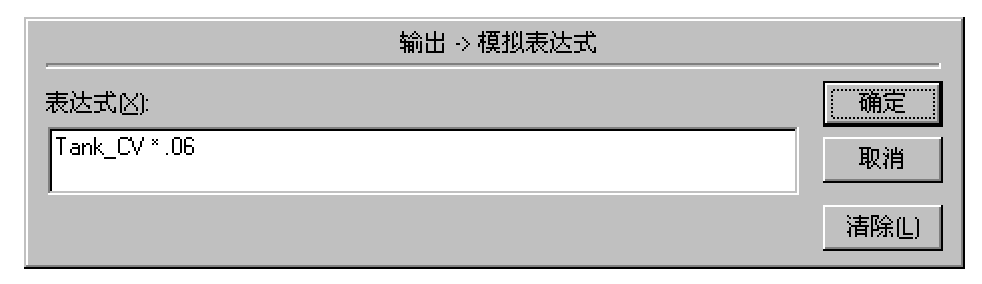
在这个表达式中， **.OnMsg** 和 **.OffMsg** 字符串将从 InTouch “标记名字典” 内离散型标记名 Pump 的定义中提取。

要创建模拟值显示链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击该对象，然后单击**动画链接**。

2. 在**值显示**区，单击**模拟**。此时会出现**输出量 -> 模拟表达式**对话框。



3. 在**表达式**框中，输入模拟型（整型或实型）标记名或等于模拟值的表达式。（您也可以在这个表达式中使用离散型标记名。它会简单地显示为 1 或 0）。

提示 右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个“触发器”标记名用作 QuickFunction 的参数，以强制更新“动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对“动画链接”的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅 第 8 章“在 InTouch 中创建 QuickScript”。

4. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在“标记名字典”中定义（远程标记名除外），则系统会提示您立即去定义它。

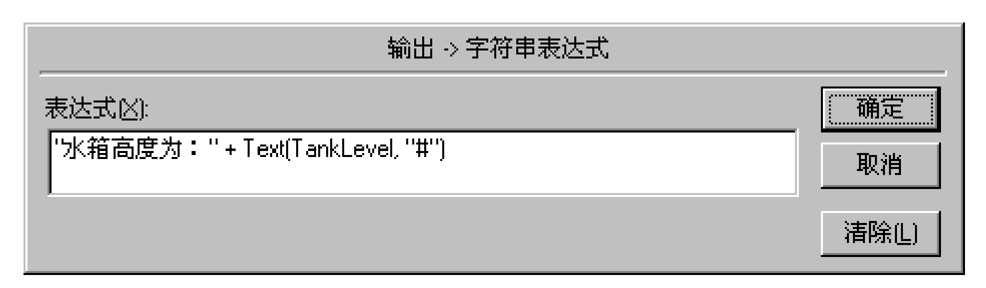
如需有关设置模拟显示对象格式的详细信息，请参阅第 2 章“使用 WindowMaker”。

要创建字符串值显示链接

1. 双击对象，或选择它然后在**特别**菜单上单击**动画链接**。此时会出现链接选择对话框。

提示 要快速访问动画链接选择对话框，请右击对象，然后再单击**动画链接**。

2. 在**值显示区**，单击**字符串**。此时会出现**输出量 -> 字符串表达式**对话框。



3. 在**表达式**框中，输入消息型标记名或等于消息型标记名的表达式。

提示 在上面的表达式中，函数 **Text()** 用于将整型标记名 TankLevel 的值转换成一个字符串。

右击**表达式**框，以访问可应用于所选文本的各个命令。

备注 您最多可以给表达式输入 256 个字符。如果要使用更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

您必须将一个 “触发器” 标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接” 的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

如需有关 QuickFunction 的详细信息，请参阅 第 8 章 “在 InTouch 中创建 QuickScript”。

4. 单击**确定**将链接附加到对象，并返回动画链接对话框。如果需要，现在您可以给该对象添加其它链接。

备注 如果输入的标记名未在 “标记名字典” 中定义 （远程标记名除外），则系统会提示您立即去定义它。

第 8 章

在 InTouch 中创建 QuickScript

编写 InTouch 脚本是 InTouch 应用程序最强大的功能之一。 InTouch QuickScript 功能可供您在满足指定条件的情况下执行特定的命令和逻辑运算。例如，键被按下、窗口被打开，值发生改变等。

QuickFunction 是您创建的且可以从其它脚本和动画链接表达式中调用的脚本。可重复使用的代码存储在一个脚本中并将它放到某个位置，从而支持在一个编辑会话中更新所有脚本的实例。

通过使用脚本，您可以创建各种自定义和自动化的系统函数。

目录

InTouch QuickScript

所有的 InTouch QuickScript 都是事件驱动的。事件可以是数据改变、条件、鼠标单击、定时器，等等。处理顺序取决于具体的应用程序。尽管相同事件触发的多个脚本在计划方式上可能存在一定的固有顺序，但系统并不保证会按任何特定的顺序执行。因此，不应就处理顺序建立任何依赖性。

下面简要介绍您可以创建的脚本类型：

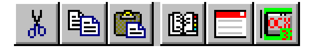
脚本类型	描述
应用程序	链接到整个应用程序。
窗口	链接到特定的窗口。
键	链接到键盘上特定的键或键组合。
条件	链接到离散型标记名或表达式。
数据改变	仅链接到标记名和 （或）**标记名点域**。
QuickFunction	您创建的且可依从其它 InTouch QuickScript 或动画链接表达式中进行调用的脚本。QuickFunction 既可以是同步也可以是异步的，而所有其它脚本类型只能是同步的。
动作按钮	与链接到**触动链接 - 动作按钮**的对象关联。
ActiveX 事件	在运行时执行 ActiveX 控件事件。
向导	运行时访问向导属性以便获取更强的功能。

使用 InTouch QuickScript 编辑器

对于所有类型的 QuickScript， InTouch QuickScript 编辑器基本上都是相同的。因此为避免赘述，本节将介绍它的公用函数和功能。各类 QuickScript 脚本特有的项目将在本章稍后部分该类 QuickScript 脚本对应的章节中作介绍。



QuickScript 编辑器工具栏



“QuickScript 编辑器”工具栏可供快速访问编辑器的各项功能：

工具	描述
	单击此按钮可以从 QuickScript 剪切所选的文本。
	单击此按钮可以从 QuickScript 复制所选的文本。
	单击此按钮可以将文本从剪贴板粘贴到 QuickScript。

工具	描述
	单击此按钮可以打开 “标记名字典”。
	单击此按钮可以从当前应用程序插入窗口名。
	单击此按钮可以打开 “ActiveX 控件浏览器”。

QuickScript 编辑器一般操作程序

本节介绍在各种 InTouch QuickScript 编辑器对话框中编写脚本时会采用的一般操作程序。某种脚本类型特有的操作程序将在本章稍后部分该类脚本对应的章节中作介绍。

QuickScript 编辑器的底部包含文本、逻辑和数学运算符按钮，您可以单击它们，以便将显示的关键字、函数或符号快速插入脚本中的光标位置。

要缩进 / 取消缩进脚本中的文本

将光标放到要缩进的行的开始位置，然后按 TAB 键。要取消缩进，请在按住 TAB 键的同时按下 SHIFT 键。

要新建脚本

在**脚本**菜单上，单击**新建**。

备注 **应用程序、窗口脚本**或**触动按钮动作**脚本没有**脚本**菜单。

此外，如果您通过在 “应用程序浏览器” 中双击脚本将它打开以便编辑，则不允许使用新建脚本功能。要新建脚本，请关闭当前打开的脚本，然后在**特别**菜单上，指向**脚本**，然后选择要创建的脚本类型。

QuickFunction、“条件” 脚本和 “数据改变” 脚本在创建时使用的是缺省名称。因此，在将另一脚本加载到编辑器之前，您必须保存 （或按 <Esc> 取消）当前显示的脚本。

创建 QuickScript 时，不能声明与先前定义的标记名同名的局部变量。

要从应用程序中删除脚本

选择要删除的文本，然后在**脚本**菜单上，单击**清除**。此时该脚本会从应用程序中完全删除。

备注 被删除的文本不会写入 Windows “剪贴板”。

要撤消上一个操作

在**编辑**菜单上，单击**撤销**。此时上次进行的编辑操作 （如粘贴）会被撤消。

提示 要快速执行此命令，请右击脚本窗口，然后单击**撤消**。除非执行的是可以逆转的操作，否则**撤消**命令不会被激活。

要选择整个脚本

在**编辑**菜单上，单击**全部选定**。此时会选定整个脚本。

提示 要快速执行此命令，请右击脚本窗口，然后单击**全部选定**。现在您便可以复制、剪切或删除整个脚本。

要从脚本剪切选定的文本

选择要删除的文本，然后在**编辑**菜单上，单击**剪切**。此时剪切的文本会从脚本中删除并被复制到 Windows “剪贴板”。现在您既可以将剪切下来的文本粘贴到脚本中的另一个位置，也可以将它粘贴到另一个脚本中。

提示 要快速执行此命令，请右击脚本窗口，然后单击**剪切**。除非选择了要剪切的文本，否则**剪切**命令不会被激活。

要从脚本复制选定的文本

选择要复制的文本，然后在**编辑**菜单上，单击**复制**。此时所复制的文本将被写入 Windows “剪贴板”。现在您便可以将所复制的文本粘贴到脚本中的另一个位置，或将它粘贴到另一个脚本中。

提示 要快速执行此命令，请右击脚本窗口，然后单击**复制**。除非选定了要复制的文本，否则**复制**命令不会被激活。

备注 剪切或复制文本时，它会自动写入 Windows “剪贴板”。在执行后续的剪切或复制命令之前，此信息会一直保留在 “剪贴板”上。

要将文本粘贴到脚本中

在**编辑**菜单上，单击**粘贴**。此时 Windows “剪贴板”的内容会被粘贴到脚本中的光标位置。

提示 要快速执行**粘贴**命令，请右击脚本窗口，然后单击**粘贴**。（如果 Windows “剪贴板”中没有内容可供粘贴，则 **粘贴**命令不会被激活）。

要清除脚本中的文本

在**编辑**菜单上，单击**清除**。此时脚本中的所有文本均会被清除。不过，该脚本并不会从应用程序中删除。如果选择此命令，然后关闭脚本编辑器并重新打开它，则该脚本会重新出现。如果只想删除脚本的部分内容，请辉亮显示要删除的文本，右击脚本窗口的空白区，然后单击**删除**。

提示 要完全删除脚本，您必须使用**脚本**菜单上的**清除**命令；或者是，选择整个脚本，右击脚本窗口的空白区，然后单击**删除**。

要将函数插入脚本

1. 在**插入**菜单上，指向**函数**，然后单击函数类别的名称。此时会出现相应的**选择函数**对话框。
2. 单击要使用的函数。此时对话框关闭，函数将自动插入脚本中的光标位置。

   可供使用的函数类型有：

函数	描述
全部	此时会出现 “选择函数”对话框，显示所有可用的函数，包括安装的各个附加程序（Recipe Manager、 SPC Pro 和 SQL Access Manager 或从 KBCD 安装的函数）的函数。 **提示** 您也可以单击 “函数”组中的 “全部”按钮来访问这些函数。
字符串	此时会出现**选择函数**对话框，显示所有可用的字符串函数。 **提示** 您也可以单击**函数**组中的**字符串**按钮来访问这些函数。
数学	此时会出现**选择函数**对话框，显示所有可用的数学函数。 **提示** 您也可以单击**函数**组中的**数学**按钮来访问这些函数。
系统	此时会出现**选择函数**对话框，显示所有可用的系统函数。例如，启动和 （或）激活另一个应用程序、读取和 （或）写入文件与磁盘信息的函数等。 **提示** 您也可以单击**函数**组中的**系统**按钮来访问这些函数。

函数	描述
附加程序	此时会出现**选择函数**对话框，显示已安装的各个附加程序 （Recipe Manager、 SPC Pro 和 SQL Access Manager）的所有可用函数。 **提示** 您也可以单击**函数**组中的**附件**按钮来访问这些函数。
其它	此时会出现**选择函数**对话框，显示所有可用的其它函数。例如，报警、历史趋势、窗口控件、 ActiveX 控件等对象的函数。 **提示** 您也可以单击**函数**组中的**其它**按钮来访问这些函数。
帮助	此时会出现**选择功能来为 ... 获得帮助**对话框，列出所有可用的函数。单击某个函数可打开它对应的 “帮助”主题。 **提示** 您也可以单击**函数**组中的**帮助**按钮来访问这些函数。
QuickFunction	此时会出现**选择函数**对话框，列出所有可从当前脚本调用的 QuickFunction 的名称。 **提示** 您也可以单击**函数**组中的**快速**按钮来访问所有的 QuickFunction。

如需有关各个脚本函数的详细信息，请参阅 “脚本函数”。

要将标记名插入脚本

1. 在**插入**菜单上，单击**标记名**。此时会出现 “标记浏览器”，并处于无限选择模式。

   提示 通过 “标记浏览器”访问的最后一个标记源中定义的标记名会显示出来。要更改标记源，请单击**标记源**箭头，并从列表中选择一个不同的标记源。

   单击**定义标记源**，以便在**标记源**列表中添加或删除标记源。

2. 双击要使用的标记名，或选择标记名，然后单击**确定**。此时 “标记浏览器”将关闭，该标记名会自动插入 Quick Script 中的光标位置。

提示 要快速访问“标记浏览器”，请双击 QuickScript 窗口中的空白区。

要访问“标记名字典”中特定标记名的定义，请在 QuickScript 窗口中输入标记名，然后双击它。

如需有关“标记浏览器”的详细信息，请参阅第 6 章“标记名字典”。

要将标记名点域插入脚本

1. 在**插入**菜单上，单击**标记名**。此时会出现“标记浏览器”，并处于无限选择模式。

提示 通过“标记浏览器”访问的最后一个标记源中定义的标记名会显示出来。要更改标记源，请单击**标记源**箭头，并从列表中选择一个不同的标记源。

单击**定义标记源**，以便在**标记源**列表中添加或删除标记源。

2. 选择要使用的标记名，然后单击**点域**箭头。从列表中选择要给标记名使用的**点域**。
3. 单击**确定**。所选的标记名**点域**将插入 QuickScript 中的光标位置。

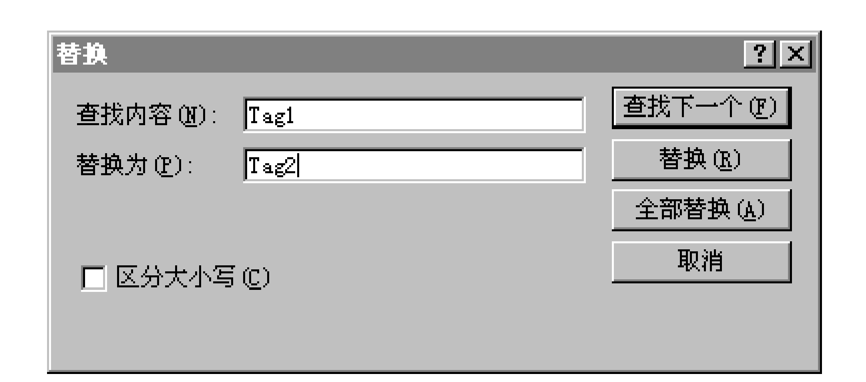
提示 要快速插入标记名**点域**，请输入后跟句点 (**.**) 的标记名，然后双击句点右侧。此时会出现**选择域名**对话框。单击要使用的**点域**。此时该对话框会关闭，所选的**点域**将自动插入 QuickScript 中的光标位置。

如需有关“标记浏览器”的详细信息，请参阅第 6 章“标记名字典”。

如需有关标记名**点域**的详细信息，请参阅联机 *InTouch 参考指南*。

要查找或替换脚本中的标记名

1. 在**编辑**菜单上，单击**查找**。此时会出现**替换**对话框。



2. 在**查找内容**框中，输入要查找 （或替换）的标记名，然后单击**查找下一个**。
3. 在**替换为**框中，输入用于替换旧标记名的新标记名，然后单击**替换**或**全部替换**。

   提示 如果只想替换旧标记名的某些实例，请单击**查找下一个**。此时 **InTouch** 将开始搜索脚本中的旧标记名。找到旧标记名时，它会被辉亮显示。单击**替换**将它替换为新标记名，或单击**查找下一个**以跳过它并继续搜索。如果要替换特定标记名的所有实例，请在搜索过程中随时单击**全部替换**。

4. 如果要查找特定大小写的标记名实例，请选择**区分大小写**选项。
5. 单击**取消**以关闭对话框。

要将窗口名插入脚本

1. 在**插入**菜单上，单击**窗口**。此时会出现**窗口名插入**对话框，并显示应用程序中所有窗口的名称。
2. 单击要使用的窗口名。此时该对话框将关闭，窗口名会自动插入脚本中的光标位置。

要验证脚本

在编写脚本时，可随时单击**验证**来检查脚本语法是否正确。

提示 单击**确定**或**保存**时，验证过程会自动执行。如果系统在验证脚本时遇到错误，则会显示一个相应的错误消息框。

如需有关脚本错误的详细信息，请参阅 “脚本编辑器错误消息”。

要保存脚本

如果要编写多个脚本，则在完成其中一个之后，您可以单击**保存**来保存它，然后在**脚本**菜单上，单击**新建**来编写另一个新脚本。

备注 **应用程序**和**窗口**脚本不支持此项功能。否则，在单击**确定**按钮时，会自动执行保存功能。

提示 如果系统在保存脚本时遇到错误，则会出现一个相应的错误消息框。

要恢复脚本

如果在更改脚本之后决定要清除所作的更改并恢复原始脚本，请单击**恢复**。

备注 在保存之后，您便不能再恢复脚本了。**应用程序**和**窗口**脚本不支持此项功能。

要退出脚本编辑器

在**脚本**菜单上，单击**编辑**。此时脚本编辑器将关闭，除非遇到错误，否则该脚本会保存。在**应用程序脚本编辑器**中，指向**文件**，然后单击**退出**。

提示 在完成脚本的编写工作之后，您也可以单击**确定**来关闭脚本编辑器。

备注 选择**退出**、**确定**，或单击对话框右上角的 **X** 按钮时，系统会自动验证脚本的正确性。

如需有关脚本错误的详细信息，请参阅 “脚本编辑器错误消息”。

要指定脚本的执行频率

在**当运行期间 / 在显示期间 / 在向下期间每 0 毫秒**框中，输入在脚本执行前等待的毫秒数。

提示 创建 “应用程序”的**在运行期间**脚本、窗口的**在显示期间**脚本、“条件”的**为真 / 假期间**脚本，或者 “键”和 “触动按钮动作”的**按下期间**脚本时，必须指定它们的执行频率 （以毫秒为单位）。

备注 WindowViewer 会按指定的时间，尽快运行这些类型的脚本。不过，并不能保证其性能良好。此外，脚本的运行速度不得快于在配置 WindowViewer 属性时指定的**计时间隔**设置。

在 Windows NT 操作系统中，脚本的执行频率不得快于 10 毫秒；在 Windows 2000 （或更高版本的）操作系统中不得快于 50 毫秒。

如需有关 “计时间隔”设置的详细信息，请参阅第 2 章 “使用 WindowMaker”。

应用程序脚本

“应用程序脚本”是与整个应用程序链接的。您可以使用应用程序脚本来启动其它一些应用程序、创建过程模拟以及计算变量等。您可以在应用程序中使用下面三种类型的 “应用程序脚本”：

应用程序脚本	描述
启动时	在应用程序最初启动时执行一次。

在运行期间	当应用程序正在运行时按指定的频率连续执行。
关闭时	在应用程序退出时执行一次。

备注 “应用程序”**启动时** QuickScript 会在打开任何窗口或发生任何运行时初始化工作之前执行。因此，您不能在 “应用程序”**启动时** QuickScript 中引用 ActiveX 方法、属性或事件。

与此类似， I/O 通讯会在 “应用程序”**启动时** QuickScript 执行之后初始化。因此，您不能在 “应用程序”**启动时** QuickScript 中引用 I/O 型标记名或远程标记名引用。此外， I/O 型标记名和远程标记名引用不会在 “应用程序”**关闭时** QuickScript 中更新。

“数据改变”或 “条件” QuickScript 不能从 “应用程序”**启动时** QuickScript 中执行。

此外，您不能使用 “应用程序”**关闭时** QuickScript 来启动其它应用程序。

要访问应用程序脚本编辑器

在**特别**菜单上，指向**脚本**，然后单击**应用程序脚本**，或者**脚本**下的 “应用程序浏览器”中，双击**应用程序**。此时会出现**应用程序脚本**编辑器。

在**脚本**下的“应用程序浏览器”中，您也可以右击**应用程序**，然后单击**打开**。



选择**在运行期间**脚本时，**每 0 毫秒**框会被激活。在框中，输入在脚本执行之前等待的毫秒数。如果要立即执行脚本，请创建一个相同的**启动时**脚本。不过，只要符合**在运行期间**脚本的条件或事件，该脚本便会以指定的频率重复执行。

窗口脚本

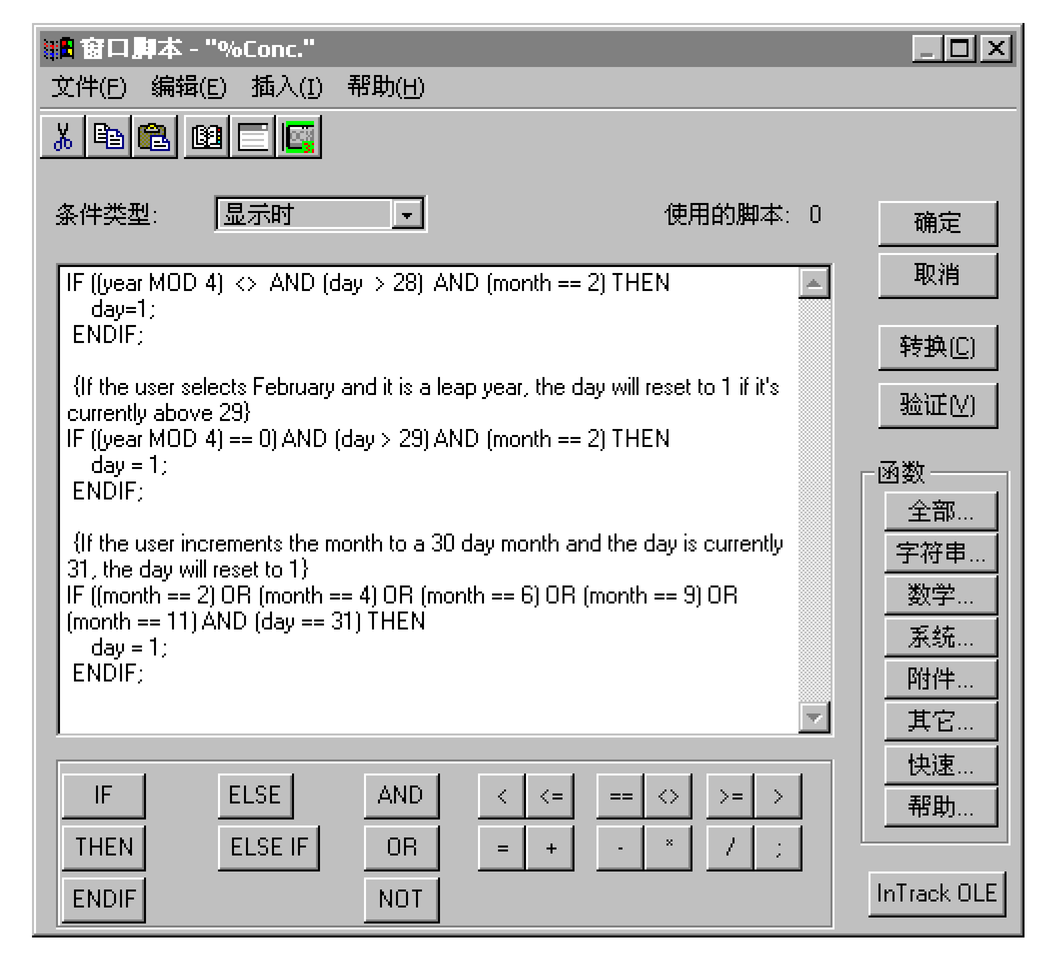
窗口脚本与特定的窗口链接。您可以给窗口使用下列三种类型的脚本：

窗口脚本	描述
显示时	在窗口最初显示时执行一次。
在显示期间	在窗口显示期间按指定的频率连续执行。
隐藏时	在窗口隐藏时执行一次。

要创建窗口脚本

在**特别**菜单上，指向**脚本**，然后单击**窗口脚本**。此时会出现**窗口脚本**编辑器。

提示 要快速访问特定窗口的 “窗口脚本”编辑器，请在**窗口**下的 “应用程序浏览器”中，右击该窗口名，然后单击**窗口脚本**。您也可以右击打开的窗口的空白区，然后单击**窗口脚本**。如果所选窗口已存在脚本，则会显示它。



选择**在显示期间**时，**每 0 毫秒**框会被激活。在框中，输入在脚本执行之前等待的毫秒数。如果要立即执行该脚本，请创建一个相同的**显示时**脚本。不过，只要符合**在显示期间**脚本的条件或事件，则该脚本便会以指定的频率重复执行。

备注 如果给活动窗口附加一个 “窗口脚本”，然后创建一个新窗口，则活动窗口的脚本可以复制到新窗口。此时会出现一个消息对话框，询问是否要复制脚本。

键脚本

“键脚本”链接到键盘上特定的键或键组合。您可以使用它们为应用程序创建全局键。例如，返回主菜单窗口、注销操作员等。您可以给键应用三种类型的 “键脚本”：

键脚本	描述
按下时	键刚按下时执行一次。
按下期间	按住键的过程中，将按指定的频率连续执行。
放开时	在键被释放时执行一次。

要访问键脚本编辑器

在**特别**菜单上，指向**脚本**，然后单击**键脚本**，或者在**脚本**下的 “应用程序浏览器”中，双击**键**。此时会出现**键脚本**编辑器。

提示 在**脚本**下的“应用程序浏览器”中，右击**键**，然后单击**打开**。



选择**按下期间**时，**每 0 毫秒**框会被激活。在框中，输入在脚本执行之前等待的毫秒数。如果要立即执行该脚本，请创建一个相同的**按下时**脚本。不过，只要符合**按下期间**脚本的条件或事件，该脚本便以指定的频率重复执行。

如需有关给脚本指定键的详细信息，请参阅 “将等价键指定给脚本”。

备注 “触动按钮动作”脚本的本地活动窗口中使用的等价键会改写使用相同等价键的全局 “键脚本”。

在 Windows 2000 中，包含 ALT 键的 SendKey QuickFunction 必须写入“放开时”脚本中。“按下时”脚本不会处理 ALT 键。

触动按钮动作脚本

“触动按钮动作脚本”类似于 “键脚本”，不同之处在于它们与链接到**触动链接 - 动作按钮**的对象关联。（脚本编辑器可以通过动画链接选择对话框进行访问）。操作员单击或者按下指定给链接的对象或按钮时，则会执行它们。您可以给对象使用三种类型的 “触动动作脚本”：

触动按钮动作脚本	描述
按下时	键刚按下时执行一次。
按下期间	在键被按住期间按照指定的频率连续执行。
放开时	键被释放时执行一次。

要创建动作按钮脚本

1. 绘制要链接到脚本的对象或按钮。
2. 双击该对象，或选择它，然后在**特别**菜单上，单击**动画链接**。此时会出现动画链接选择对话框。

提示 要快速访问对话框，请右击对象，然后单击**动画链接**。

3. 在**触动按钮**区，单击**动作**。此时会出现 **InTouch -> 动作脚本**编辑器。



选择**按下期间**时，**每 0 毫秒**框会被激活。在框中，输入在脚本执行之前等待的毫秒数。如果要立即执行该脚本，请创建一个相同的**按下时**脚本。不过，只要符合**按下期间**脚本的条件或事件，该脚本便以指定的频率重复执行。

如需有关给脚本指定键的详细信息，请参阅 “将等价键指定给脚本”。

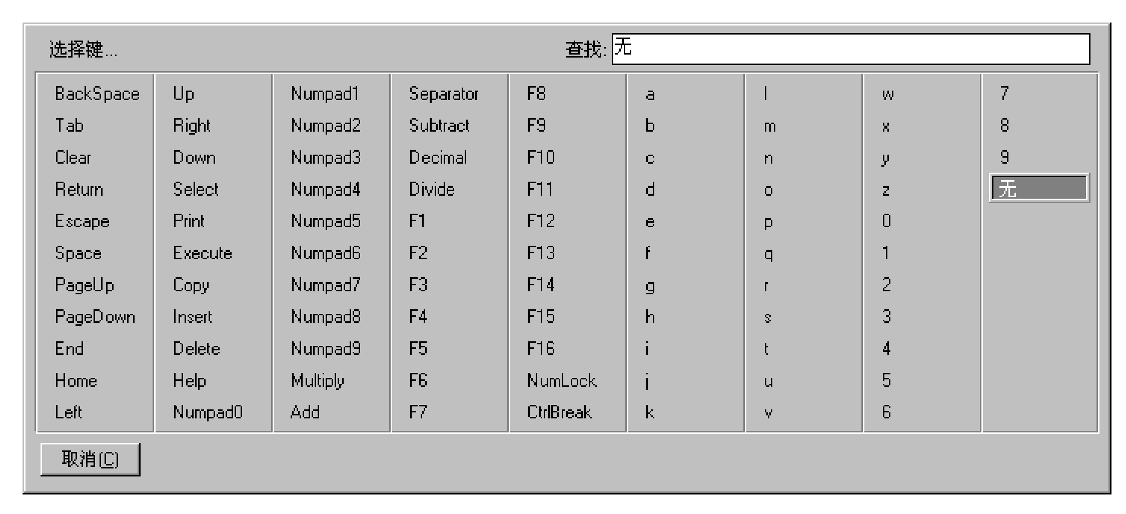
备注 “触动按钮动作”脚本的本地活动窗口中使用的等价键会改写使用相同等价键的全局 “键脚本”。此外，仅当包含该对象的窗口处于活动状态时，等价键才会被激活。

将等价键指定给脚本

“键脚本”和 “触动动作脚本”编辑器与其它 QuickScript 编辑器略有不同。因为要创建的脚本将应用于键，所以您还必须指定操作员需按下以执行该脚本的键。

要将键指定给键脚本

1. 如果希望操作员在按下该键以执行脚本的同时还必须按住 **Ctrl** 和（或） **Shift** 键，请选择 **Ctrl** 和 （或） **Shift**。
2. 单击**键**以选择要指定给脚本的键。此时会出现**选择键**对话框。



3. 单击所需的键。此时对话框会自动关闭，您选择的值会自动输入**键**框。

条件脚本

“条件脚本”既可以与离散型标记名链接，也可以与一个等于真或假的表达式链接。您也可以使用包含模拟型标记名 （见下例）的离散表达式。您可以给条件应用四种类型的脚本：

条件脚本	描述
为真时	条件变为真时执行一次。
为假时	条件变为假时执行一次。
为真期间	条件为真时连续执行。
为假期间	条件为假时连续执行。

要访问条件脚本编辑器

1. 在**特别**菜单上，指向**脚本**，然后单击**条件脚本**，或者在**脚本**下的“应用程序浏览器”中，双击**条件**。此时会出现**条件脚本**编辑器。

提示 在脚本下的“应用程序浏览器”中，您还可以右击**条件**，然后单击**打开**。



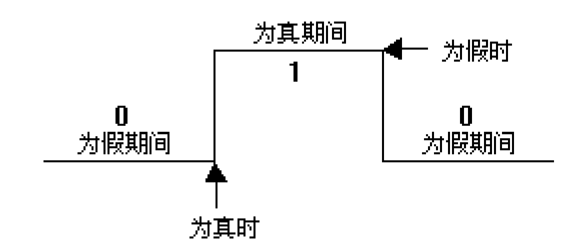
2. 由于“条件脚本”根据是否满足条件来执行，因此必须在**条件**框中指定条件（离散型标记名或表达式）。

备注 每个条件只能应用一个脚本。

提示 您也可以使用等价于为真或假的模拟型标记名的离散表达式。例如， TankLevel >= 75。在本例中，在标记名 TankLevel 的值大于或等于 75 时，脚本便会执行。

在脚本执行之前，条件的值必须变为真或假。例如，如果在 WindowViewer 启动时的初始值为真，则只有该值变为假然后再变为真，才会执行**为真时**脚本。

您可以给相同的条件应用所有这四种类型的脚本。在经过指定毫秒数之后，**为真期间**和**为假期间**脚本均将开始执行。要立即执行，请创建一个相同的**为真时**或**为假时**脚本。例如：



3. 在**注释**框中，输入有关脚本的任何其它注释。

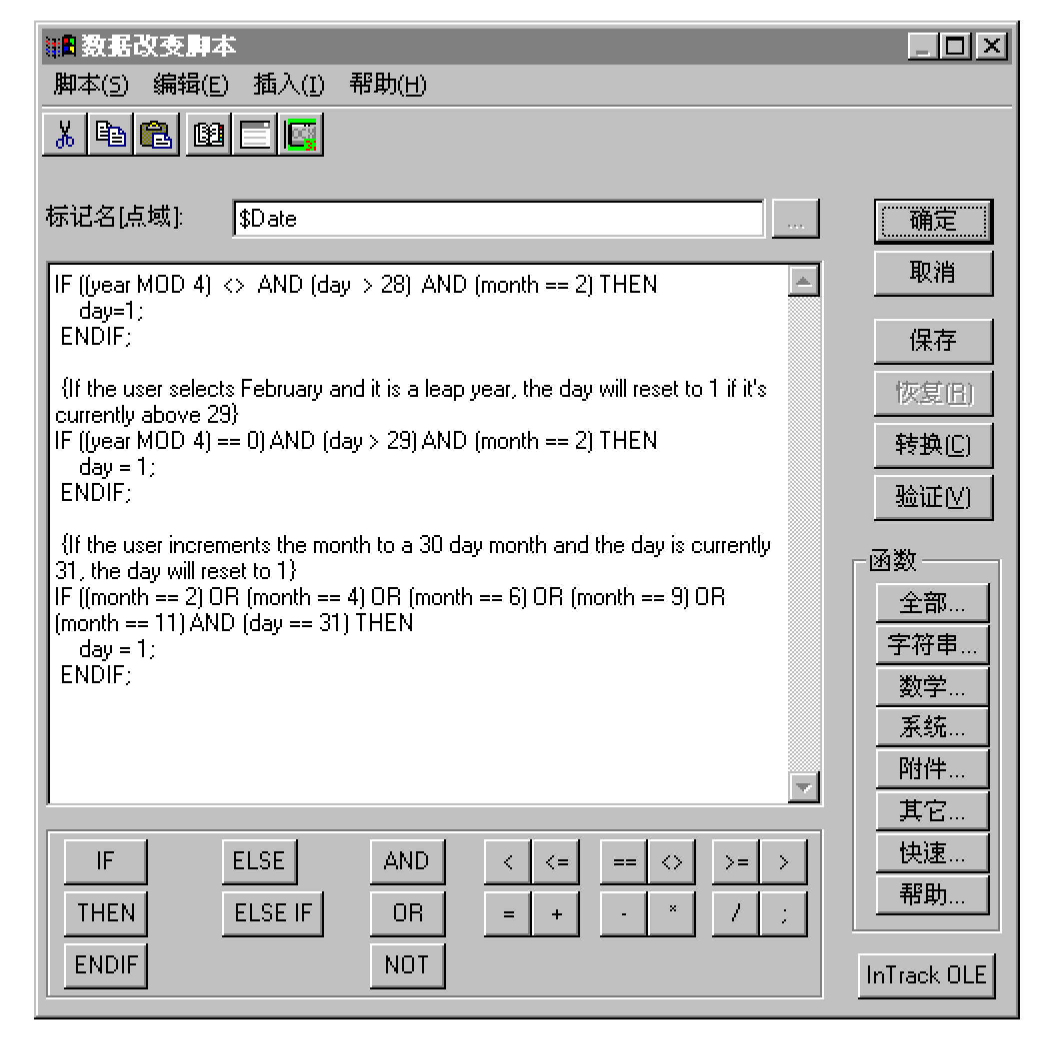
数据改变脚本

"数据改变脚本"仅与标记名和（或）**标记名点域**链接。在标记名或**标记名点域**的变化值大于在"标记名字典"中给该标记名定义的死区值时，它们会执行一次。

要访问数据改变脚本编辑器

1. 在**特别**菜单上，指向**脚本**，然后单击**数据改变脚本**，或者在**脚本**下的"应用程序浏览器"中，双击**数据改变**。此时会出现**数据改变脚本编辑器**。

提示 在脚本下的“应用程序浏览器”中，您也可以右击**数据改变**，然后单击**打开**。



2. 因为“数据改变脚本”是根据数据值的变化来执行的，所以必须在**标记名 [点域]** 框中指定一个标记名或**标记名点域**。

3. 在**插入**菜单上，单击**标记名**，或双击脚本窗口。此时会出现**选择标记**对话框。

   要选择一个不含**点域**的标记名，请双击该标记名，或选择它然后单击**确定**。此时所选标记名将插入脚本中的光标位置。

提示 要快速访问**选择标记**对话框，请双击脚本编辑窗口中的空白区。要在“标记名字典”中访问特定标记名的定义，请输入该标记名，然后双击它。

要选择**点域**，请首先选择要使用的标记名，然后单击**点域**箭头，并选择要与所选标记名关联的**点域**。单击**确定**。此时所选标记名**点域**将插入脚本中的光标位置。

提示 要快速插入标记名**点域**，请在标记名后面输入句点 (.)，然后双击句点右侧。此时会出现**选择域名**对话框。单击要使用的**点域**。此时该对话框将关闭，所选**点域**将自动插入脚本中的光标位置。

如需有关标记名**点域**的详细信息，请参阅联机 *InTouch 参考指南*。

如需有关“标记浏览器”的详细信息，请参阅第 6 章“标记名字典”。

注意！“条件脚本”或“数据改变脚本”中修改（或写入）的标记名不得用作“数据改变脚本”或条件脚本表达式的标记名。

例如：Script1 调用 Script2， Script2 调用 Script3，而 Script3 则试图调用 Script1（仍处于活动状态）。此时，最后一个调用将不会执行。为避免出现此种情形，不要将“条件”脚本或“数据改变”脚本中修改的标记名用作“数据改变”脚本或“条件”脚本表达式中的标记名。

ActiveX 事件脚本

大多数 ActiveX 控件都有与之关联的事件。例如，单击、双击、按下鼠标和按键都是许多 ActiveX 控件中使用的典型事件。为此，系统提供了一些 InTouch“ActiveX 事件”脚本来支持这些事件动作。您可以给每个事件关联一个“ActiveX 事件”脚本。 ActiveX 控件事件将在运行时 (WindowViewer) 执行。

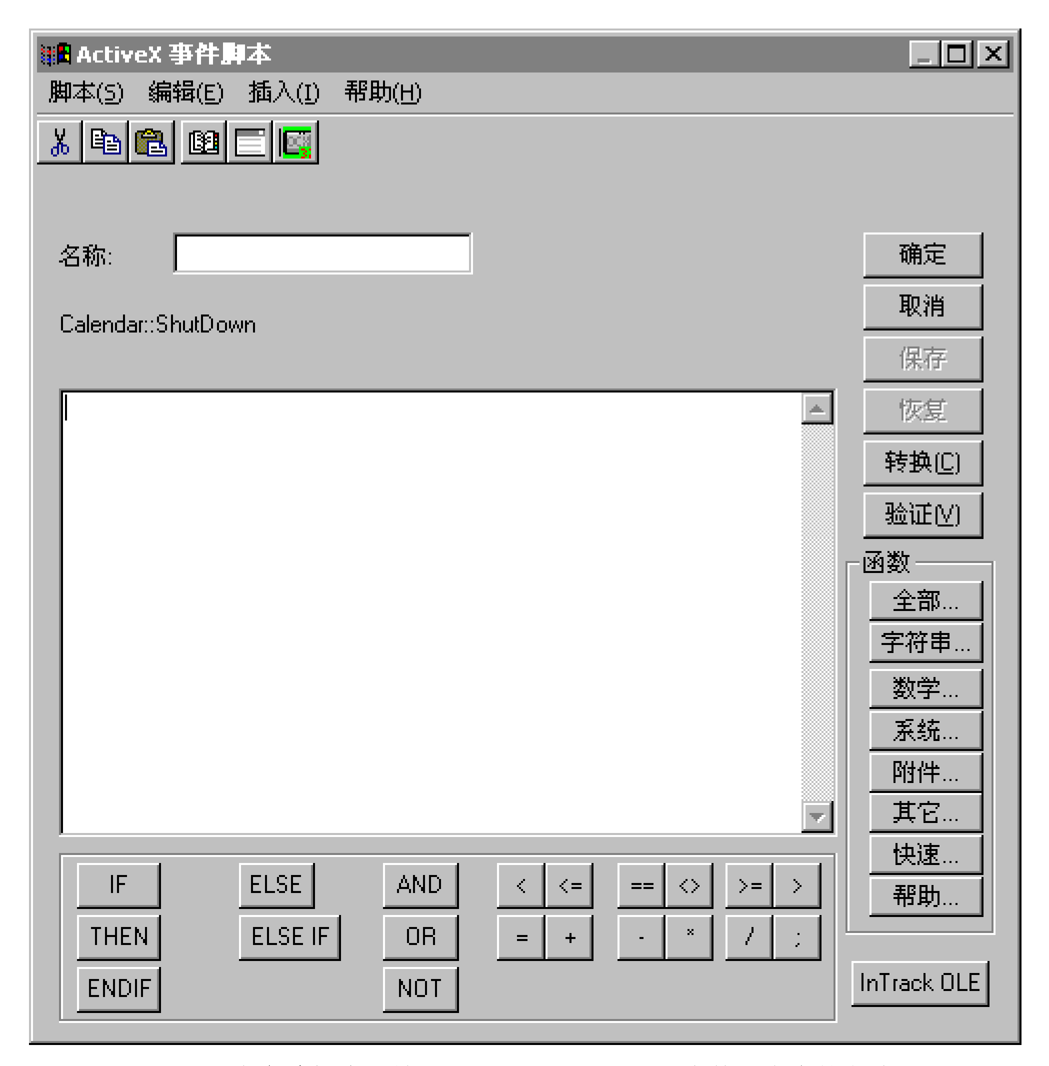
要访问 ActiveX 事件脚本编辑器

1. 单击 ActiveX 控件**属性**对话框中的**事件**标签，以激活**事件**属性页。例如：



2. 双击**脚本**列中的空白单元，或者输入 "ActiveX 事件" 脚本的名称，然后单击**确定**。

3. 如果当前不存在使用所输入名称的 "ActiveX 事件" 脚本，则会出现一个消息框，询问是否要立即去创建它。单击**确定**。此时会出现 **ActiveX 事件脚本**编辑器。



4. 在**名称**框中，输入用于识别 “ActiveX 事件”脚本的名称。
5. ActiveX 控件方法类似于 ActiveX 控件属性。属性是与对象关联的数据成员，而方法则是控制对象的数据函数。要访问 ActiveX 控件方法，请在**插入**菜单上单击 **ActiveX**。此时会出现 **ActiveX 控件浏览器**。



ActiveX 控件浏览器会显示应用程序中使用的所有 ActiveX 控件的名称。选择某个控件的名称时，会显示它对应的方法。选择要插入脚本的方法，然后单击**确定**。

QuickFunction

QuickFunction 指可以在其它脚本或表达式中进行编写和调用的脚本。它们存储于创建它们的应用程序中。通过从其它脚本或表达式调用 QuickFunction，您可以一次性创建脚本，然后再多次重复使用它。使用这些脚本可以减少脚本中来回复制的重复代码量，从而降低应用程序的维护工作量。可重复使用的代码存储在一个脚本中，并被放入一个位置，因此您可以在一个编辑会话中更新所有的脚本实例。

QuickFunction 可定义为异步模式，因此执行期间它们会在主 WindowViewer （运行时）进程的后台运行。此功能使得 WindowViewer 可以将 SQL 数据库调用之类耗时的操作同主程序流分开。需执行这些耗时的操作时，通过创建一个异步 QuickFunction，您便可提供了一种方法，使得在执行这些操作的同时，仍可以让所有的动画链接及其它 InTouch 功能保持活动状态。

例如，假设您要在 for next 循环中执行计算。通过使用异步 QuickScript，您就可以在执行计算的同时而又不停止 InTouch 系统的执行。随后，您可以使用 **IsAnyAsyncFunctionBusy()** 函数来确定计算何时完成，并向操作员显示计算结果。

```
If IsAnyAsyncFunctionBusy(120) ==1 then  Show "Calculation
   running Window";  Endif;
```

备注 我们不建议在异步 QuickScript 中运行 SQL 命令，否则如果任何 SQL 命令出现故障，均可能会导致系统不稳定。

如需有关异步脚本的详细信息，请参阅异步 QuickFunction 脚本。

提示 动态链接表达式框最多可使用 255 个字符。不过，您可以创建一个更为复杂的 QuickFunction，然后从动画链接表达式框中调用它。这样一来，您便可以使用 **CALL** 语句来调用包含 **RETURN** 语句的复杂脚本，该语句可以将结果返回给表达式。

例如，将各包含 30 个字符的多个标记名 （使用 "**&**" 和 "**:**"）添加到一起时，您只能在表达式中使用 8 个标记名 （加上空格）。不过，通过在表达式中使用语句 **CALL MYSCRIPT($SECOND)**，您就可以执行包含数百个 30 字符长标记名的 QuickFunction。此 QuickFunction 使用 RETURN 语句给表达式提供一个返回值。

备注 您必须将一个 “触发器”标记名用作 QuickFunction 的参数，以强制更新 “动画链接”。例如，将标记名 $Second 用作 QuickFunction 的参数时，可以在每次 $Second 值发生改变时对 “动画链接”的表达式求值，从而每隔一秒调用该 QuickFunction 一次。

QuickFunction RETURN 语句不能用于返回消息型或字符串型值。

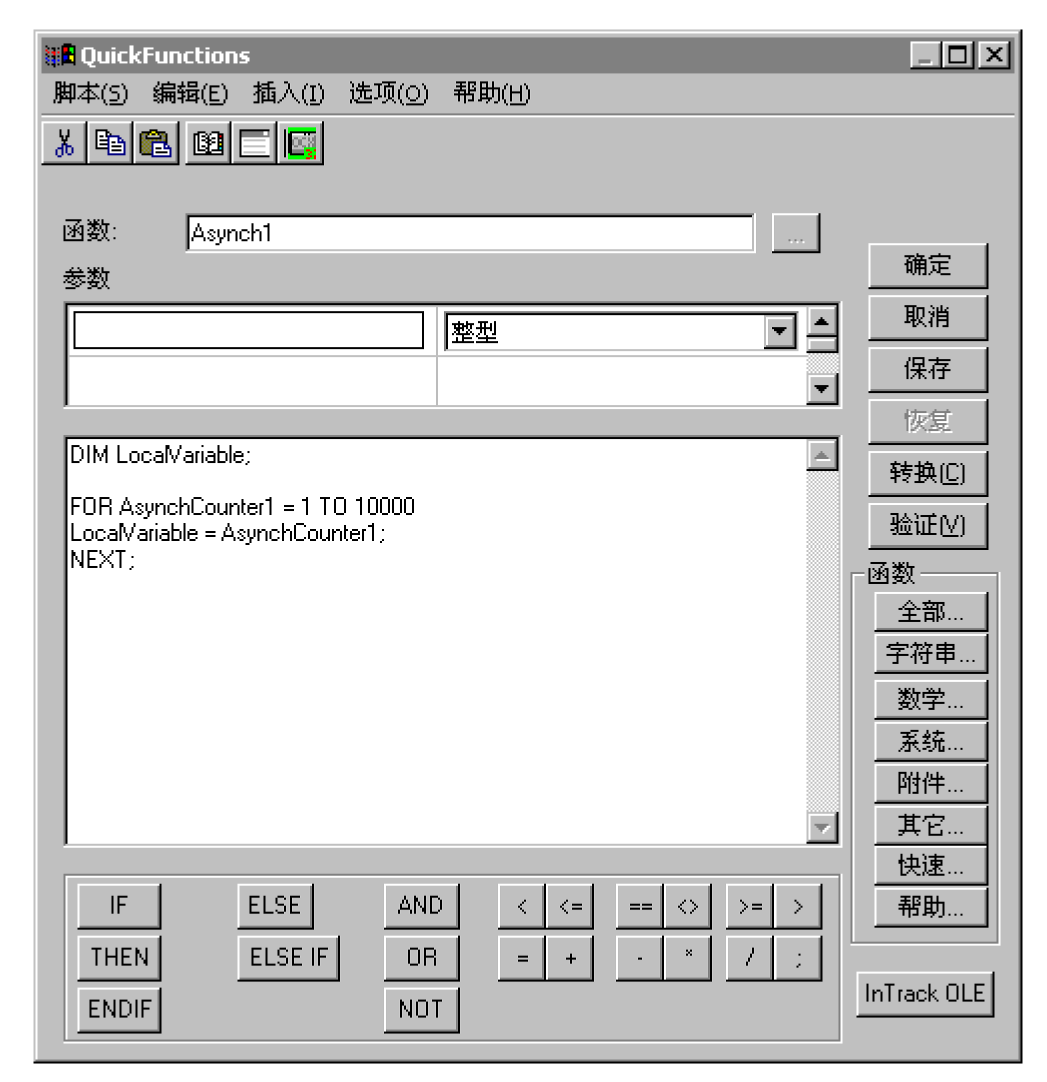
在创建并保存 QuickFunction 之后，您就可以立即从任何其它脚本或表达式中调用它的名称。

如需有关 QuickFunction 语法的详细信息，请参阅联机 *InTouch 参考指南*。

要创建 QuickFunction

1. 在**特别**菜单上，指向**脚本**，然后单击 **QuickFunction**，或者在**脚本**下的 “应用程序浏览器”中，双击 **QuickFunction**。此时会出现 **QuickFunction** 脚本编辑器。

提示 在脚本下的“应用程序浏览器”中，您也可以右击 **QuickFunction**，然后单击**打开**。



2. 在**函数**框中，输入 QuickFunction 的名称。

提示 该名称最长可达 31 个字符。（不能使用空格和重复的名称）。这是其它 QuickScript 或表达式用于调用该 QuickFunction 的名称。在 QuickScript 编辑器中单击**全部**或**快速**按钮，或者在**插入**菜单的**函数**下，单击**全部**或 **快速函数**命令，此名称也会出现在**选择函数**对话框中。

3. 在**参数**框中，输入 QuickScript 每个参数的名称，然后单击参数的箭头，并从列表中选择它的数据类型。

   有效的数据类型包括：

数据类型	描述
Integer	用于传递整型变量、标记名或常数值。
Real	用于传递实型变量、标记名或常数值。
Discrete	用于传递离散型变量、标记名或常数值。
Message	用于传递字符串变量、标记名或常数值。

   下面是保留关键字，不得用作参数名：

Return	**Call**	**Dim**	**As**	**RetVal**
Integer	**Real**	**Discrete**	**Message**	

   参数名是局部变量，仅存在于定义它们的 QuickFunction 中。每个 QuickFunction 最多可以使用 16 个参数。参数名最长可达 31 个字符，并且不得使用空格。此外，参数名还必须以希腊字符 (A-Z) 开头。不能使用重复的名称。

   不要将标记名用作参数名。标记名的优先级大于同名的参数名，因此您的 QuickScript 可能会无法正确执行。参数名被视作局部变量，因此它不占用标记名计数。

4. 在输入参数名并选择它们的类型之后，就可以开始编写 QuickFunction 了。

QuickFunction 参数表达式

脚本参数是通过值来传递的。参数表达式可以是返回整型、实型、离散型、消息型数据值的任何脚本表达式。在执行 QuickFunction 之前，会通过调用脚本来解析所有的参数表达式值。例如：

```
CALL Stuff (5.6, 237, "PI");
```

在这个表达式中，“实型”常数 5.6 用作参数 1，“整型”常数 237 用作参数 2，“消息型”常数 "PI" 用作参数 3。

```
CALL Temp (IntegerTag);
```

将 IntegerTag 当作参数表达式值进行传递。

```
CALL ValveOpen (Tag.MaxEU -5);
```

将计算所得的值 (Tag.MaxEU -5) 当作参数表达式值进行传递。

参数数据类型匹配

调用语句参数表中的数据类型与被调用的已保存 QuickFunction 的数据类型之间必须保持严格的左右对应关系。参数的数目也必须与已保存的 QuickFunction 参数表中的参数数目完全相同。必要时会强制执行类型转换，将 "实型" 值转换成 "整型" 以及将 "整型" 值转换成 "实型"。通过使用这种修改值类型的功能，您可以将任意模拟型参数传递给任何其它模拟类型。

例如，如果您将 "实型" 值 1.23 传递给一个 "整型" 参数，则它将只使用 1， .23 会被丢弃。与此类似，如果将 "整型" 值 1 传递给一个 "实型" 参数，它会将 1 更改为 1.0。不过，尽管有了这种强制转换的功能，我们仍建议让调用参数类型与相应的 QuickFunction 参数表严格匹配。

有效的 **QuickFunction** 语法

QuickFunction 会返回一个值。 QuickFunction 语句的语法和形式如下：

CALL*QuickFunctionName ([arg1, ... arg16]);*

其中：

CALL	是所有 QuickScript 和表达式调用 QuickFunction 所需的关键字。
QuickFunctionName	是 1 到 31 个字符组成的字符串，对应于指定给已保存 QuickFunction 的名称 。
([arg1, ...arg16])	是以括号括起的 0 到 16 个逗点分隔参数表达式。

如需有关 QuickFunction 语法的详细信息，请参阅联机 *InTouch 参考指南*。

使用 **Return** 语句

QuickFunction 是可以从另一个脚本 （调用脚本）调用的脚本。下面是一个调用 QuickFunction 的脚本的示例：

```
RETURNRESULT = CALL MyFunction( StartHour, EndHour);
```

QuickFunction 使用 **RETURN** 语句强制将值 "返回" 给调用脚本。遇到 **RETURN** 时， QuickFunction 便立即结束执行。在此时， QuickFunction 会将一个值返回给调用脚本。返回的数据类型可以是离散型、整型或实型。

在上例中，标记名 **RETURNRESULT** 必须是一个接收传递值的离散型、整型或实型标记名。遇到 **RETURN** 语句时， QuickFunction 将不会继续执行。

在下例中，如果标记名 ConvertType 等于 0，则会计算表达式 abs((Max / Maxprogress) * 100)，并将此实型数据结果返回给调用它的脚本。

```
IF ConvertType == 0 THEN
    RETURN abs( (Max / Maxprogress) * 100 );
ELSE
    RETURN 0;
ENDIF;
```

返回值的数据类型由上下文确定。例如：

```
RETURN AnalogTag;
```

如果 “标记名” **AnalogTag** 定义为 “内存整型”，则 **RETURN** 语句会返回一个模拟值给调用位置。只能返回一个值。

异步 QuickFunction 脚本

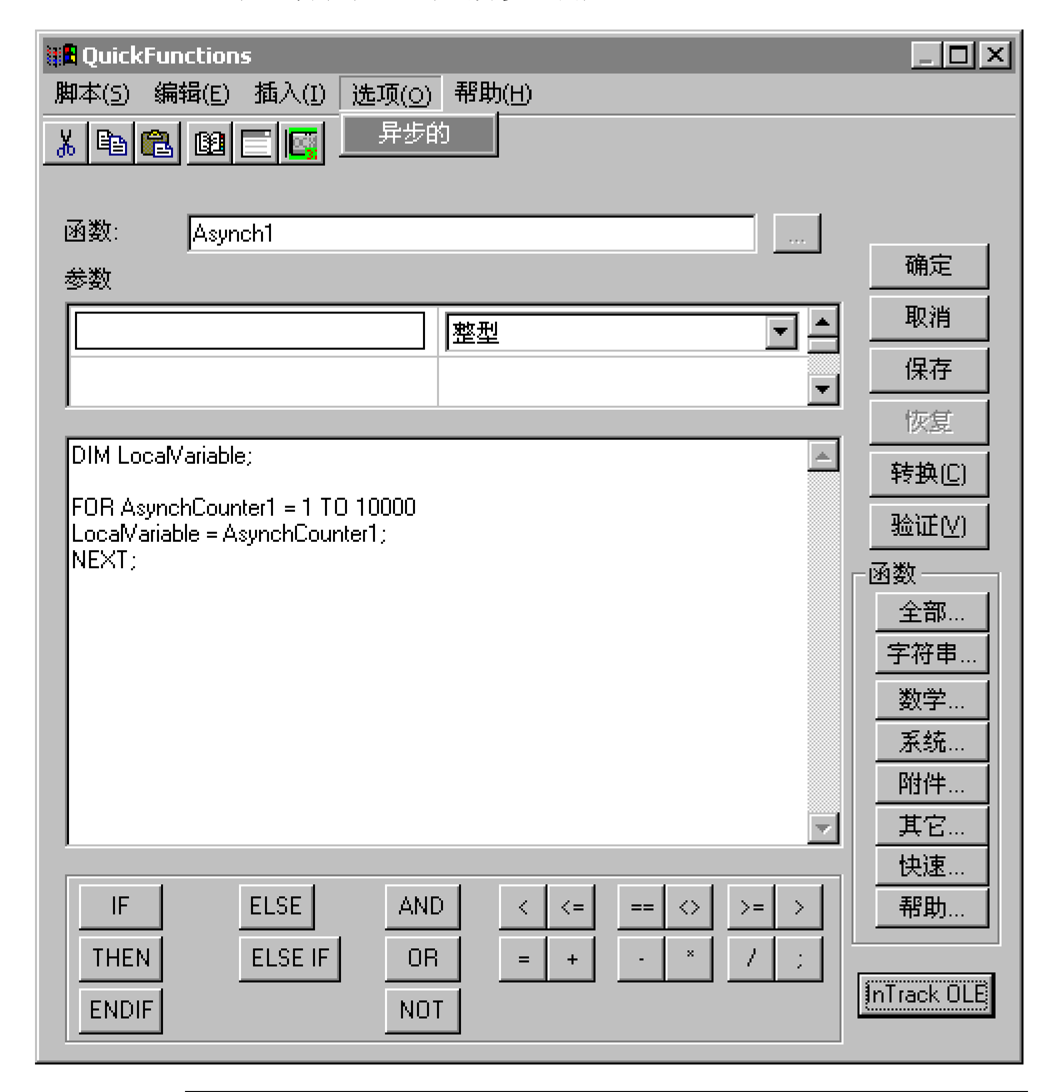
您可以将 QuickFunction 定义为异步模式。（QuickFunction 是唯一支持异步功能的 InTouch 脚本类型）。 WindowViewer 遇到调用异步 QuickFunction 的情况时，便会产生一个单独的线程。在产生新的线程之后， WindowViewer 便可以自由地继续调用更多的脚本 （包括更多的异步 QuickFunction）、等待该异步脚本结束，或更新图形窗口。异步 QuickFunction 实际上是从新产生的线程中启动的。在异步脚本执行完毕之后，新产生的线程便告结束。线程机制对于操作员而言是透明的。

备注 异步 QuickFunction 不返回任何值。因此，异步 QuickFunctions 不能用在动画链接表达式中。此外，您可以同时执行的异步 QuickFunction 数目不受限制。不过，为确保系统性能良好，强烈推荐不要同时执行三个以上的异步 QuickFunction。此外，不能同时运行同一个异步 QuickFunction 的多个实例。

要创建异步 QuickFunction

1. 创建 QuickFunction。

2. 在**选项**菜单上，单击**异步**。例如：



备注 在运行时，异步 QuickFunction 在开始执行之后便无法停止。不过，如果操作员停止运行所有脚本（通过**逻辑**菜单上的**停止逻辑**命令，或按下链接到 **$LogicRunning** 系统标记名的按钮），则不会开始执行新的异步 QuickFunction。

控制异步脚本

您可以使用 **IsAnyAsyncFunctionBusy()** 函数来确认是否有异步 QuickFunction 正在运行。通过使用此函数，您可以让调用异步 QuickFunction 的 QuickScript 等待所有其它异步 QuickFunction 先完成处理。这样该 QuickScript 便能自我重新同步。

此函数的有效语法是：

***DiscreteTag* = IsAnyAsyncFunctionBusy(*timeout*);**

其中：

DiscreteTag	是一个离散类型标记名，给它返回的值如下： 如果函数发生超时，则等候所有正在执行的 QuickFunction 完成，并将 1 (true) 返回给 *DiscreteTag*。 如果没有异步 QuickFunction 正在运行，则立即返回 0 (false)，否则 QuickFunction 将等待发生超时。如果发生超时后没有异步 QuickFunction 正在运行，则它也会返回 0。
timeout	是一个表示等待秒数的整型值，用于确定是否有任何异步 QuickFunction 正在运行。

例如，假定要使用异步 QuickFunction 连接若干个 SQL 数据库，并且您知道需要两分钟才能建立连接。首先，您需要执行异步 QuickFunction 来连接 SQL 数据库。接着，您将启动函数 **IsAnyAsynchFunctionBusy(120)**，让 SQL 有足够时间在该 QuickFunction 完成之前建立这些连接。

不过，如果在 2 分钟之后这些连接仍未建立，且异步 QuickFunctions 仍忙于建立连接，则函数 **IsAnyAsynchFunctionBusy()** 会返回 1 (true)。现在您就可以显示一则错误信息，告诉操作员 SQL 连接失败。

例如，您可以使用下面的窗口**显示时** QuickScript：

```
IF IsAnyAsyncFunctionBusy(120) == 1 THEN
   SHOW "SQL Connection Error Dialog";
ENDIF;
```

使用局部变量

您可以在脚本中声明局部变量来存储临时结果，并使用临时脚本值来创建复杂的计算，如此既不影响或减少授权标记名计数，又可提高系统性能。

在相同的脚本中，局部变量或标记名可以互换使用。不过，在脚本运行结束之后，局部变量便会失去它们的值及存在意义，但其中的标记名是全局性的，因此可以保留它们的值。与标记名不同，局部变量只能在脚本主体中声明。在给定的脚本主体中可以声明的局部脚本变量数目仅受可用内存的限制。声明局部变量之后，您就可以在相同脚本主体中的一个或多个表达式中使用它。在脚本主体中添加局部变量名的表达式和语法规则与标记名的相同，唯一的例外是，局部变量不支持**点域**引用。

与标记名类似，局部变量同时可用在语句和表达式的左边或右边，这些语句和表达式可以包括不同数据类型的其它局部变量和标记名。

备注 您不能定义与现有标记名同名的局部变量。您也不能创建与已定义的局部变量同名的标记名。如果发生此种情形，则局部变量的优先级会高于标记名，因此标记名值会反映对局部变量所作的任何改变。

有效的局部变量语法

每个局部变量均必须在脚本中用一个单独的 **DIM** 语句进行声明。（一行一个变量，并且不允许层迭）。 **DIM** 语句的语法和格式如下：

```
DIM LocalVarName [ AS data-type ];
```

其中：

DIM 是必需的关键字。

LocalVarName 是符合标记名格式和限制的变量名。变量名最长可达 32 个字符，并且必须以 A-Z 或 a-z 开头。其余的字符可以是：A-Z、a-z、0-9、!、@、-、?、#、$、%、 _、 \ 以及 &。

如果脚本中已声明的变量名和标记名之间发生抵触（二者同名），则变量名将优先于标记名。例如，假设数据库中定义了一个标记名 "Temp"，并且您声明 "**DIM** *Temp* **AS** *Integer*"。在进行声明的脚本中，语句中使用 "Temp" 的表达式将引用与局部变量 "Temp" 而非标记名 "Temp" 关联的值。

AS 是一个可选关键字。

如果在 **DIM** 语句中省略了 **AS** 语句，则缺省条件下，变量会声明为整型数据类型。例如：

```
DIM LocVar1;
```

等同于：

```
DIM LocVar1 AS Integer;
```

data-type 它可以是下面的四个关键字之一：

Integer
DIM *LocVar1* AS *Integer*;

Real
DIM *LocVar2* AS *Real*;

Discrete
DIM *LocVar3* AS *Discrete*;

Message
DIM *LocVar4* **AS** *Message*;

InTouch **DIM** 语句不能层迭。例如，下面的示例是无效的，因此无法使用：

```
DIM LocVar1 AS Integer, LocVar2 AS Real;
DIM LocVar3, LocVar4, LocVar5, AS Message;
```

要在 InTouch 中声明多个变量，则必须为每个变量输入一个单独的 DIM 语句。例如，下面的示例是有效的：

```
DIM LocVar1 AS Integer;
DIM LocVar2 AS Real;
```

备注 数据类型关键字不区分大小写。

DIM 语句行必须用分号 (;) 作结束。

层迭 **DIM** 语句不受支持。

DIM 语句可位于脚本主体中的任何位置。但它必须在第一个引用它的脚本语句或表达式之前。

如果在 **DIM** 语句之前引用局部变量，则验证脚本时，脚本编辑器会认为它是一个标记名，并要求您去定义它。

如需有关局部变量语法的详细信息，请参阅联机 *InTouch 参考指南*。

创建 FOR-NEXT 循环脚本

FOR-NEXT 循环用于在脚本的单次执行过程中多次执行脚本中的某个函数 （或一组函数）。 FOR-NEXT 循环的一般格式如下：

```
FOR AnalogTag = start_expression TO end_expression [STEP
    change_expression]
    ...statements...
IF (condition) THEN
    [EXIT FOR;]
    ENDIF;
    ...statements...
```

```
NEXT;
```

其中：

[ ]	括号中的项目为可选参数。
粗体	**大写**粗体项目表示脚本语言的保留关键字。
斜体	*斜体*的小写项目表示变量数据。
AnalogTag	InTouch 模拟型标记名。
start_expression	有效的 InTouch 表达式，用于初始化 *AnalogTag* 的值以执行循环。
end_expression	有效的 InTouch 表达式，如果 *AnalogTag* 大于 *end_expression*，则脚本的执行会立即跳至 **NEXT** 语句之后的语句。（这仅适用于递增循环的情形，如果是递减循环，则在 IntegerTag 小于 *end_expression* 时，循环将会终止）。
change_expression	有效的 InTouch 表达式，用于定义在执行 **NEXT** 语句之后 *AnalgTag* 的递增量或递减量。 **备注** *Change_expression* 既可以是正数，也可以是负数。如果 *Change_expression* 为正，则 *Start_expression* 必须小于或等于 *end_expression*，否则不会执行循环中的语句。如果 *Change_expression* 为负，则 *Start_expression* 必须大于或等于 *end_expression*，否则不会执行循环主体。如果未设置 **STEP**，则 *Change_expression* 缺省为 1。
...statements...	一个或多个有效的 InTouch 脚本语言语句。这些语句可以是嵌套的 **FOR** 循环。嵌套循环要求使用不同于外部循环的 *change_expressions*。
FOR	标志 "For" 循环的开始。
TO	标志 *end_expression* 的开始。
STEP	标志 *change_expression* 的开始。
EXIT FOR	立刻终止循环，同时脚本跳转至 **NEXT** 语句后的第一个语句执行。
NEXT	标志循环语句的结束。

执行 FOR...LOOP 函数时， InTouch：

1. 将 AnalogTag 设置为等于 start_expression。
2. 检查 AnalogTag 是否大于 end_expression。若确实如此，则 1InTouch 会退出循环。（如果 change_expression 为负，则 InTouch 将检查 AnalogTag 是否小于 end_expression）。
3. 执行语句。
4. 按增量 1 或指定的 change_expression 去增加 AnalogTag。

5. 重复步骤 2 到 4。

嵌套 FOR-NEXT 循环

FOR-NEXT 循环可以嵌套。可能的嵌套层数取决于系统的内存和可用资源。

屏幕更新和性能影响

在执行 FOR-NEXT 循环的过程中， InTouch 中的屏幕更新子系统会暂时停止，直至循环结束。

- 在执行 FOR-NEXT 循环的过程中，屏幕上所有的动画均会暂停。因此，您不能使用 FOR-NEXT 循环给对象设置在屏幕上运动的动画效果，所有的移动只有在循环结束后才会发生。
- 实时趋势会暂停。
- “正在更新”的历史趋势会暂停。
- 在循环执行过程中，屏幕上的显示值不会更新。尽管这些变量中可能存在新的值，但只有在循环完成后这些新值才会显示出来。
- FOR-NEXT 循环主体中修改过的 I/O 型标记名值只有在循环执行结束后才会传递给 I/O 服务器。因此，如果在 FOR-NEXT 循环的每次交互期间修改 I/O 型标记名的值，则只有该 I/O 型标记名的最终值会传递给 PLC。

备注 FOR-NEXT 循环会暂停 InTouch 中的所有操作。在执行期间，WindowViewer 中不会有数据移进和移出、动画链接更新，也不执行其它脚本，包括异步 QuickFunction。不过，异步 QuickFunction 中使用的 FOR-NEXT 循环不会暂停其它运算。

循环执行时间限制

缺省条件下， FOR-NEXT 循环必须在 5 秒钟内完成执行。这是 FOR-NEXT 子系统内置的一种安全限制。所有的 FOR-NEXT 循环均会强制使用该时间限制。例如：

```
FOR X = 1 TO 1000000
   FileWriteMessage("C:\LOG.TXT","Hello");
NEXT;
```

备注 通过在应用程序目录下的 INTOUCH.INI 文件中添加下列开关，可以扩展此限制。

LoopTimeout=20

其中， 20 是提前终止循环之前的秒数。

此循环很可能会超过 5 秒的限制。在 Wonderware Logger 中，会出现一则消息，具体如下：

```
95/03/07 07:34:40.550/VIEW    /
   Exceeded loop time limit of 5 seconds.
95/03/07 07:34:40.550/VIEW    /
   FOR-NEXT Timeout at X = 65464
```

此错误消息指出在满足 *end_condition* 之前 FOR-NEXT 循环已经终止，它还提供了循环终止时的循环变量值。此信息可供跟踪哪一条 FOR-NEXT 语句存在问题。

备注 5 秒时间限制仅当 FOR-NEXT 循环中每次到达 NEXT 语句时计算。例如，假定您要执行下面的脚本：

```
FOR Index = 1 to 10

SQLInsert(ConnectionID,"ORG","list1");
SQLInsert(ConnectionID,"ORG","list2");
SQLInsert(ConnectionID,"ORG","list3");
SQLInsert(ConnectionID,"ORG","list4");

NEXT;
```

如果每个 **SQLInsert( )** 要花 12 秒完成，则在退出循环之前，所有 4 个插入均会执行完毕；因为只有在执行到 NEXT 语句时才会去估计是否超出 5 秒限制。

循环执行后的循环变量值

像是在 Visual Basic （以及其它大多数流行的 Basic 编程语言）中那样，循环结束时循环变量的值定义如下：

从初始值 *Start_Condition* 开始， Index 值每次增加 *Step_Expression*，直至到达最后一次循环，此时 Index 值超过 *End_Expression* 值。

因此，如果某循环是：

```
FOR Index = 2 TO 25 STEP 7
   { Some script statements }
NEXT;
```

则 “索引”值将如下变化：

迭代	值	计算
1	2	
2	9	2 + 7
3	16	2 + 7 + 7
4	23	2 + 7 + 7 + 7
5	30	2 + 7 + 7 + 7 + 7

在该点，当值到达 30 时，由于超过了 *End_Expression*，循环将会停止。Index 最终的值是 30。

嵌套控制结构

控制结构可以置于其它控制结构中（如 FOR...NEXT 循环内的 IF...THEN 语句）。一个控制结构放到另一个控制结构中也就是所谓的*嵌套*。

示例：

```
FOR TagX = 1 TO 5

FOR TagY = 1 TO 10
    ...statements...
    IF (condition) THEN
    [EXIT FOR;]
    ENDIF;
    ...statements...
    NEXT;

NEXT;
```

其中：

第一个 NEXT 结束内层的 FOR 循环，最后一个 NEXT 结束外层的 FOR 循环。与此类似，在嵌套的 IF 语句中， ENDIF 语句会自动应用于前面最近的 IF 语句。

退出控制结构

EXIT FOR 语句可供您从 FOR 循环中直接退出。就语法而言， EXIT FOR 语句很简单。 EXIT FOR 可以根据需要在 FOR 循环内出现多次。例如：

```
FOR TagX = 1 TO 10;

...statements...

IF (condition) THEN
    EXIT FOR;

ENDIF;

...statements...

NEXT;
```

下面是各种 FOR-NEXT 循环脚本的一些例子：

例 1 “简单数学运算 2”

此循环通过设置标记名 NumberToRaise 和 Power 的值输入链接，允许配置 “乘方数”和 “幂”：

```
Product = 1;

NumberToRaise = 4;

Power = 12;

FOR Index = 1 TO Power
    Product = Product * NumberToRaise;

NEXT;
```

上述脚本执行完毕之后， Product 的值会成为 16,777,216。

例 2 “使用间接标记名的复杂 **FOR-NEXT**”

此循环利用 FOR-NEXT 结构的 "EXIT FOR" 和 " STEP" 部分在 100 个标记名上进行搜索，以找到等于 NumberEntered 的标记名。

备注 本示例假定已经存在 100 个**内存整型**标记名 (TAG1 - TAG100)。操作员将数字输入标记名 NumberEntered 中，此循环将搜索 TAG1 - TAG100 以找出匹配值。此外，还创建了一个间接模拟型标记名：*IndirectTag*

```
Found = 0;
FOR Index = 1 TO 100
    IndirectTag.NAME = "TAGNAME" + TEXT( Index, "#" );
    IF (IndirectTag.NAME == ("TAGNAME"+
        Text(NumberEntered,"#"))) THEN
        Found = 1;
        EXIT FOR;
    ENDIF;
NEXT;
IF (Found==1) THEN
    Show "NumberFound"; {window notifying search was
        successful}
ELSE
    Show "NumberNotFound";
ENDIF;
```

在脚本执行完毕之后，将会显示一个窗口，指出是否找到了该数值。

备注 请注意此脚本中两个附加函数的用法：**Show()** 和 **TEXT()**。

示例 **3**

此循环执行奇数运算，但演示了嵌套 FOR-NEXT 循环的用法，以及 FOR-NEXT 结构中 "STEP" 部分的用法：

```
MyTag = -1;
FOR Index = 1000 TO -1000 STEP -5
    IF (MyTag > Index) THEN
        FOR Index2 = 1 TO 10 STEP 2
            MyTag = MyTag * (Index + 11);
        NEXT;
    ENDIF;
NEXT;
```

该脚本处理完毕之后，这些值会是：

MyTag = -7776、 Index = -1005 以及 Index2 = 11。

脚本编辑样式和语法

InTouch 脚本编辑器支持两种 “样式”的脚本：“简单”和 “复杂”。简单脚本允许赋值、比较、执行简单的数学函数等。复杂脚本能以 IF-THEN-ELSE 语句的形式执行一些逻辑运算。此外， InTouch 还支持使用内置的复杂函数，以及本身的 QuickFunction。

一个示例函数是 **StartApp(*ApplicationName*)** 函数，此函数实际启动在参数 "(*ApplicationName*)" 中指定的 Windows 应用程序。在 “简单”和 “复杂”的脚本中，均可以使用函数。下节将完整介绍每种样式。

表达式和脚本的基本语法

QuickScript 和动画链接表达式对话框中所用的语法与计算器的代数语法相似。大多数表达式都是使用以下形式的赋值语句来编写的：

```
a = (b - c) / (2 + x) * xyz;
```

此语句将等号 (=) 右边表达式的值赋给变量 "a"。每个表达式必须以分号 (;) 结尾。表达式中的运算数可以是常量也可以是变量。单个标记名必须出现在赋值运算符 **(=)** 的左边。

内存或 **I/O 消息**型标记名可使用加号 (+) 运算符进行串联。例如，标记名就可以进行串联以用在**间接**型标记名中。如果创建了下面这样的 “数据改变脚本”，则每当 "Number" 值发生改变时，间接标记名 "Setpoint" 便会相应地发生改变：

```
Number=1;

Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

其中：结果是 "Setpoint1"。

简单脚本

简单脚本提供诸如赋值、数学运算和函数等逻辑功能。下面是这类脚本的一个例子：

```
React_temp = 150;

ResultTag = (Sample1 + Sample2)/2;

{this is a comment}

Show "Main Menu";
```

在本例中，脚本给标记名 "React_temp" 赋值 "150"。 "Sample1" 与 "Sample2" 相加，所得结果将除以 "2"，并且在脚本时运行屏幕上会显示 "Main Menu" 窗口。

备注 请注意，每个逻辑语句均必须以分号 (;) 结尾，且一个脚本可以包括多个逻辑语句。此外在脚本编辑器中允许加上注释。注释用一对花括号 {} 标明。此外，还会使用函数 **Show** 和参数 "Main Menu" (WindowName)，以便打开指定的窗口。

除简单的赋值、数学运算符和函数之外， InTouch 还支持在 “运算符” 使用多种其它 “运算”，如标记名、常数等）。下面列出的所有运算均支持**离散**、**整型**和**实型**标记名。**消息**型标记名只能用在比较和赋值运算中。下面是 InTouch 支持的运算的列表：

需要一个运算数的运算

<table>
<tr><td>~</td><td>取补</td></tr>
<tr><td>-</td><td>取反</td></tr>
<tr><td>NOT</td><td>逻辑非</td></tr>
</table>

需要两个运算数的运算

<table>
<tr><td>*</td><td>乘</td></tr>
<tr><td>/</td><td>除</td></tr>
<tr><td>+</td><td>相加和串联</td></tr>
<tr><td>-</td><td>减</td></tr>
<tr><td>=</td><td>赋值</td></tr>
<tr><td>MOD</td><td>模除</td></tr>
<tr><td>SHL</td><td>左移位</td></tr>
<tr><td>SHR</td><td>右移位</td></tr>
<tr><td>&</td><td>位与</td></tr>
<tr><td>^</td><td>异或</td></tr>
<tr><td>|</td><td>同或</td></tr>
<tr><td>**</td><td>乘方</td></tr>
<tr><td><</td><td>小于</td></tr>
<tr><td>></td><td>大于</td></tr>
<tr><td><=</td><td>小于或等于</td></tr>
<tr><td>>=</td><td>大于或等于</td></tr>
<tr><td>==</td><td>等于 （“等价于”）</td></tr>
<tr><td><></td><td>不等于</td></tr>
<tr><td>AND</td><td>逻辑与</td></tr>
<tr><td>OR</td><td>逻辑或</td></tr>
</table>

运算符的优先级

下面列出了运算符的运算优先级次序。列表中的第一个运算符会首先求值，第二个运算符其次求值，以此类推。同一行中的运算符具有相同的优先级。运算符按最高优先级到最低优先级的次序列出。

( ) 最高优先级

- , NOT, ~

**

* , /, MOD

+, -

SHL, SHR

<, >, <=, > =

==, <>

&

^

|

=

AND

OR 最低优先级

优先级示例

因为 * 的优先级高于 +，

所以 B+C*D 等价于 B+(C*D) ；

因为 * 和 / 具有相同的优先级，

所以 B / C*D 等价于 (B / C)*D ；

再看一下其它几个示例：

B* - D 等价于 B* (- D) ；

B or C and D 等价于 B or (C and D)。

运算符描述

上面列出的运算符的参数可以是数字，也可以是标记名。外加括号的参数是可选的，而且运算符名称不区分大小写。

括号 ( )

括号用于确保运算求值的顺序正确。它们也可使复杂的表达式更易于阅读。括号内的运算最先求值 （优先级高于没有括号时的其它优先级规则）。如果您不清楚或需要改写运算的优先顺序，请使用括号。在下例中，使用括号强制 B 和 C 相加后再乘以 D：

```
( B + C ) * D;
```

取反 (-)

取反是个一元运算符，它将正整数或正实数转换为负数。

取补 (~)

此运算符产生 32 位整数的补数。换句话说，它将每一个 0 位变成 1 位，每个 1 位变成 0 位。补数运算符是一个接受整型运算数的一元运算符。

乘方 (**)

此二元运算符返回第一个数字 （基数）的第二个数字 （幂）次的值。基数和幂可以是任意实数或整数，但必须遵守以下限制：

零为底和幂为负数均是无效的。

示例：**"0 ** - 2"** 和 **"0 ** -2.5"**

底为负数和幂为分数是无效的。

示例：**"-2 ** 2.5"** 和 **"-2 ** -2.5"**

运算数无效时得出的结果为 0。此外，运算结果不能过大或过小，以致无法用实数表示。示例：

```
1 ** 1 = 1.0
3 ** 2 = 9.0
10 ** 5 = 100,000.0
```

乘 (*)、除 (/)、加 (+)、减 (-)

这些二元运算符执行基本的数学运算。加号 (+) 也可用于串联**内存**或 **I/O 消息**型标记名。例如，多个标记名可以串联起来用在**间接**标记名中。如果创建了下面这样的 "数据改变脚本"，则每次 "Number" 值发生改变时，间接标记名 "Setpoint" 会相应地发生改变：

```
Number=1;
Setpoint.Name = "Setpoint" + Text(Number, "#" );
```

其中：结果会是 "Setpoint1"。

模除 (MOD)

MOD 是一个二元运算符，它用右边的整数除左边的整数。所得余数即是 MOD 运算的结果。示例：

```
97 MOD 8 等于 1
63 MOD 5 等于 3
```

左移位 (SHL)、右移位 (SHR)

SHL 和 **SHR** 是只使用整型运算数的二元运算符。运算符左边的 32 位字的二进位内容将移动 （右或左）运算符右边指定的位数。从字中移出的位会被丢弃。移走后的空位将用零填入。（移位是无符号移位）。

位与 (&)

这是一个二元位运算符，它逐位比较两个 32 位整型字。此运算符一个常见地用于是屏蔽一系列位。本例中的运算将 “屏蔽掉”（设置为 0）一个 32 位字中的高 24 位。例如：

```
result = name & 0xff;
```

异或 (^) 和同或 (|)

或是位逻辑运算符，可逐位比较两个 32 位整型字。“异**或**”寻找对应位置的相反位状态。如果对应位相同，则得出结果 0 ；如果对应值相反，则得出结果为 1。示例：

0 ^ 0 得 0

0 ^ 1 得 1

1 ^ 0 得 1

1 ^ 1 得 0

“同**或**”检查对应位是否为 1。如果两个对应位有一个为 1，结果即为 1。只有在两个对应位均为 0 时才得 0。例如：

0 | 0 得 0

0 | 1 得 1

1 | 0 得 1

1 | 1 得 1

赋值 (=)

赋值是接受整型或实型运算数的二元运算符。每个语句只能包含一个赋值运算符。赋值运算符左边只能有一个名称。赋值运算符的等号 (=) 读作 “赋值为”或 “设置为”。

备注 不要将等号与 IF-THEN-ELSE 子句及相对关系中使用的等价符号 (==）相混淆。

比较 （<、 >、 <=、 >=、 ==、 <>）

这些运算符用在 IF-THEN-ELSE 语句中，用以执行各种基于表达式状态的指令。

AND （与）、 OR （或）和 NOT （非）

这些运算符只适于离散型标记名。不过，如果这些运算符用在整数或实数上，则它们将执行如下*转换*：

实型到离散型： 如果实数是 0.0，则离散值是 0，否则为 1。

整型到离散型： 如果整数是 0，则离散值是 0，否则为 1。

因此，如果该语句是："Disc1 = Real1 AND Real2;" 并且 Real1 是 23.7、Real2 是 0.0，则 Disc1 将被赋予 0，因为 Real1 将转换成 1， Real2 会转换为 0。

复杂脚本

复杂脚本使您能以 IF-THEN-ELSE 脚本的形式执行逻辑运算

并使用 FOR-NEXT 脚本结构来处理循环。下面是 IF-THEN-ELSE 脚本的一个示例：

```
IF React_temp > 200 THEN
    React_temp_sp = 150;
    PRValve = 1;
    PlaySound("c:\alert.wav",1);
ELSE
    PRValve = 0;
    PlaySound("c:\All_Ok.wav",1);
ENDIF;
```

在本例中，脚本检查反应炉温度是否高于 "200"。如果超过，则将 "150" 赋给 "React_temp_sp" 标记名，此时 "PRValve" 将会打开，并通过调用 **Playsound()** 函数播放 "alert.wav" 文件。反之，如果反应炉温度低于 "200"，则会关闭 "PRValve" 并播放 "All_Ok.wav" 文件。

备注 请注意，每个 IF 语句都要有对应的 ENDIF 语句。同时，如果脚本的功能不需要，则不必包括 ELSE 语句。注意这个复杂脚本中函数 **PlaySound(*path_text,number*)** 的用法。

简单数学运算

此循环执行简单的迭代数学计算。执行时， Product 等于 NumberToRaise 的 10 次方，即 **Product=NumberToRaise^{10}**。

```
Product = 1;
NumberToRaise = 4;
FOR Index = 1 TO 10
    Product = Product * NumberToRaise;
NEXT;
```

在执行该脚本之后， "Product" 值会变为 "1048576"。

备注 FOR-NEXT 循环会暂停 InTouch 中所有的运算。在执行期间，WindowViewer 中不会有数据移进和移出、动画链接更新，也不执行其它脚本，包括异步 QuickFunction。不过，异步 QuickFunction 中使用的 FOR-NEXT 循环不会暂停其它运算。

脚本中的 IF-THEN-ELSE 和比较

IF-THEN-ELSE 语句用于根据表达式的状态有条件地执行各种不同的指令。下面的比较操作符用于设置 IF-THEN-ELSE 语句的条件：

< 小于

> 大于

<= 小于或等于

>= 大于或等于

== 等于 （“等价于”）

<> 不等于

下面是各种复杂脚本的一些示例：

没有 ELSE 子句的 IF-THEN 语句：

```
IF a <> 0 THEN
   a = a + 100;
ENDIF;
```

包含一个 ELSE 子句的 IF-THEN-ELSE 语句：

```
IF temp > 500 THEN
   Disc = 1;
   Real = 43.7;
ELSE
   Disc = 0;
   Real = 93.4;
ENDIF;
```

包含一个 ELSE IF 子句但无 ELSE 子句的 IF-THEN-ELSE 语句：

```
IF temp > 500 THEN
   Disc = Disc * 10;
ELSE
   IF temp > 250 THEN
      x = y / z;
      a = abc + def;
   ENDIF;
ENDIF;
```

包含一个 END IF 子句和一个 ELSE 子句的 IF-THEN-ELSE 语句：

```
IF temp > 500 THEN
   Disc = Disc - 10;
ELSE
   IF temp < 250 THEN
      Disc = Disc + 10;
```

```
    ELSE
        Disc = Disc + 50;
        Real = 100;
    ENDIF;
ENDIF;
```

备注 每个 IF 必须有一个匹配的 ENDIF，并且在每个语句行地结尾必须输入一个分号。

包含多个 ELSE IF 子句和一个 ELSE 子句的 IF-THEN-ELSE 语句：

```
IF temp > 100 THEN
    temphihi = 1
    Disc = 50;
ELSE
    IF temp > 80 THEN
        temphi = 1;
ELSE
    IF temp < 10 THEN
        templo = 1;
ELSE
    IF temp < 30 THEN
        templolo = 1;
ELSE
        tempok = 1;
    ENDIF;
ENDIF;
ENDIF;
ENDIF;
```

IF-THEN-ELSE 语句，检测 “条件 1”*或*“条件 2”：

```
IF (pump1 < 50.0) OR (pump2 < 50.0) THEN
    alarm-1 = 1;
ELSE
    alarm-1 = 0;
ENDIF;
```

IF-THEN-ELSE 语句，检测 “条件 1”*和*“条件 2”：

```
IF (pump1 < 50.0) AND (pump2 < 50.0) THEN
    alarm-2 = 1;
ELSE
    alarm-2 = 0;
ENDIF;
```

IF-THEN-ELSE 语句检测是否相等：

```
IF a > 50 THEN
    IF b == 100 THEN
        c = 0;
    ENDIF;
ENDIF;
```

导入 QuickScript

将 QuickScript 从一个 InTouch 应用程序导入当前应用程序可以节省您相当的开发时间。它同时还给您提供了一种简便、快捷的方法，供您创建远程标记名引用。它可供您重复使用先前创建的 QuickScript。将一个 InTouch 应用程序中的 QuickScripts 移到另一个应用程序时，您必须使用**文件**菜单上的**导入**命令。

如需有关远程标记名引用的详细信息，请参阅第 6 章 “标记名字典”。

要导入 QuickScript

1. 关闭当前应用程序中的所有窗口。
2. 在**文件**菜单上，单击**导入**。此时会出现**浏览文件夹**对话框。

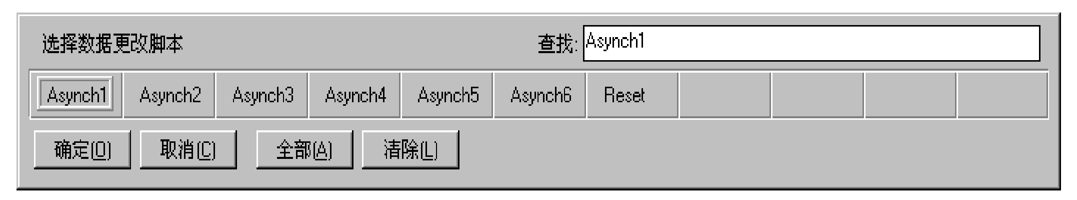


3. 找到并且选择包含要导入的 QuickScript 的应用程序目录 （文件夹）。
4. 单击**确定**。此时会出现下面的对话框。



5. 选择要导入的 QuickScript 类型。

6. 单击**选择**。此时会出现**选择 *ScriptType* 脚本**对话框。



7. 选择要导入的 QuickScript，然后单击**确定**以关闭该对话框。

备注 将 "ActiveX 事件" 脚本从一个应用程序导入另一个应用程序时，所有的 "ActiveX 事件" 脚本均会被导入。此外，为了让导入的 "ActiveX 事件" 脚本在新应用程序中能够正常运行，原先创建脚本时的相同 ActiveX 控件和相同的事件也必须用在新应用程序中，并且必须加载到内存中。如果关闭了包含 ActiveX 控件的窗口，则它的 "ActiveX 事件" 脚本，或包含与该 ActiveX 控件关联的脚本函数的任何其它 InTouch QuickScript 将不能正常执行。

8. 单击**导入**。系统会自动开始将所选的 QuickScript 导入当前应用程序。

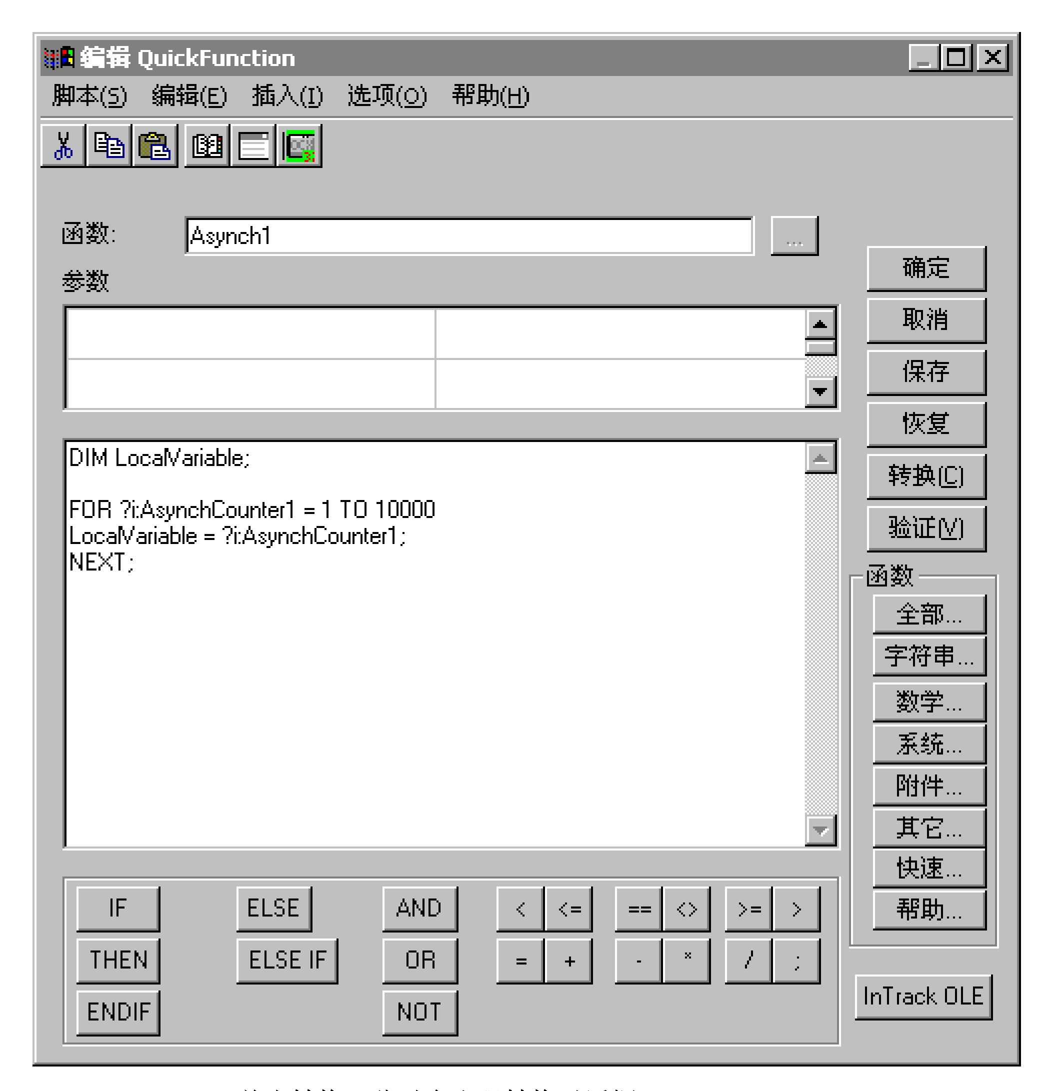
备注 将 QuickScript 导入新应用程序时， QuickScript 中所有的标记名将随同该 QuickScript 一起导入，但它们不会添加到 “标记名字典”。相反，它们会自动转换成 “占位符”标记名。要使用它们，您必须先转换这些占位符标记名，如果当前尚未在新应用程序的 “标记名字典”中定义它们，则系统会要求您去逐个定义。

导入的 QuickScript 中的标记名转换成占位符标记名时，每个标记名的开头会加上三个索引字符。例如，在导入一个离散型标记名时，该标记名的开头会加上三个字符 **?d:** 做前缀。导入长度为 30、31 或 32 个字符的标记名时，每个标记名的开头仍会添加这三个索引字符。不过，增加这三个字符不会截短现有标记名的长度。例如，仅对于占位符标记名，32 个字符标记名会增加到 35 个字符。这三个额外的空间仅分配给占位符标记名。标准标记名不支持这种标记名长度的增加。

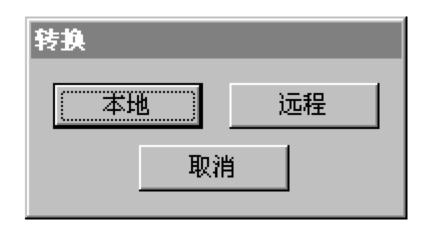
要转换导入脚本中的占位符标记名

1. 在**特别**菜单上，指向**脚本**，然后单击导入的 QuickScript 类型，或者在**脚本**下的 “应用程序浏览器”中，双击导入的 QuickScript 类型。此时会出现 QuickScript 编辑器，显示所选脚本类型的第一个 QuickScript。例如，如果导入了 **QuickFunction** 脚本，则会出现 **QuickFunction** 脚本编辑器。

提示 要快速打开导入的 QuickScript，请在 “应用程序浏览器” 中双击**脚本**，然后双击 QuickScript 类型。



2. 单击**转换**。此时会出现**转换**对话框。



3. 单击**本地**将 QuickScript 中的标记名转换为本地标记名。
4. 在转换标记名之后，单击 QuickScript 编辑器中的**确定**。

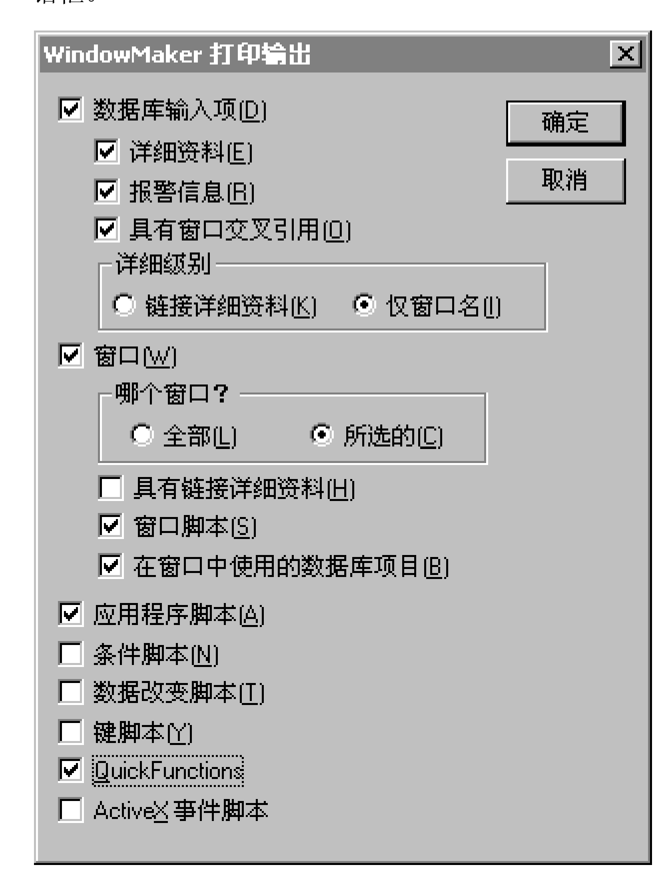
如需有关转换为远程标记名引用的详细信息，请参阅第 6 章 “标记名字典”。

打印脚本

您可以打印每种 InTouch QuickScript 类别中的所有脚本。

要打印脚本

1. 在**文件**菜单上，单击**打印**。此时会出现 **WindowMaker 打印输出**对话框。



2. 要打印窗口脚本，请选择**窗口**，然后选择**窗口脚本**。在**哪个窗口？**组中，选择**全部**以打印应用程序中所有窗口的脚本。要打印指定窗口的脚本，请选择**所选的**。此时会出现**打印窗口**对话框。选择要打印其脚本的窗口，然后单击**确定**。

提示 如果选择的窗口没有与之链接的脚本，则报表上会打印下面的消息："Window Scripts for *Window Name*:none."

---

3. 要打印某种 QuickScript 类型的所有脚本，请选择该 QuickScript 类型，然后单击**确定**。

脚本函数

InTouch 提供了大量的内置函数，利用这些函数，您可以链接到对象或按钮，或在脚本中使用它们来执行各种任务。例如，确认警报、隐藏窗口、改变当前绘制趋势图的标记名等等。

这些函数可通过**插入**菜单或单击 “脚本编辑器”**函数**区域中的相应按钮进行访问。在相应的对话框中选择函数之后，该函数及所需的参数将自动粘贴到脚本中的光标位置。在将该函数粘贴到脚本之后，您就可以辉亮显示要修改的参数，然后输入新值。

字符串函数

字符串函数用在字符串变量上。下面简要介绍各个字符串脚本函数。

函数	描述
DText	根据离散型标记名的值来更改消息型标记名。
StringASCII	返回指定消息型标记名中第一个字符的 ASCII 值。
StringChar	返回指定的 ASCII 码对应的字符。
StringFromIntg	将整型值转换成其它形式的字符串。
StringFromReal	将实数转换成浮点数或指数形式的字符串表达式。
StringFromTime	将时间值 （自 1970 年 1 月 1 日起以秒为单位）转换成特定的字符串表示法。
StringInString	返回*文本*中*搜索内容*首次出现的位置。
StringLeft	返回从文本最左端的字符开始由 Chars 指定的字符数。
StringLen	将文本长度返回给一个整型结果。
StringLower	将文本中的所有大写字符转换为小写形式，并将产生的字符串放入 MessageResult。
StringMid	从 *StartChar* 位置开始，从文本提取由 Chars 指定的字符数。此函数与它对应的函数 **StringLeft()** 和 **ShringRight()** 稍有不同，它允许用户指定要从消息型标记中提取的字符串的首尾位置。
StringReplace	替换或更改指定字符串的特定部分。

函数	描述
StringRight	返回从文本最右端的字符开始由 Chars 指定的字符数。
StringSpace	在消息型标记名或表达式中生成空格字符串。
StringTest	检查文本的第一个字符以确定它是否为特定的类型。
StringToIntg	将消息型标记名中的数值转换为可应用数学运算的整数值。
StringToReal	将消息型标记名的数值转换为可应用数学运算的实数（浮点）值。
StringTrim	从文本中移除多余的空格。
StringUpper	将文本中所有的小写字符转换成大写形式。
Text	让消息型标记名根据指定的 ***Format_Text*** 显示模拟型（整型或实型）标记名的值。

如需有关每个函数的有效语法及其用法示例的详细信息，请参阅联机 *InTouch 参考指南*。

数学函数

数学函数可用在整型或实型标记名上。在下面的数学函数中，**ResultNumericTags** 和 **InputNumericTags** 既可以是**实型**，也可以是**整型**，并且可随意混合使用。不过请记住，将函数的非整型结果返回给**整型**标记名时，会导致该结果被截短。（小数点右边的部分会被丢弃）。下例假定 ***ResultNumericTag*** 被定义为**内存实型**或 **I/O 实型**标记名。

下面简要介绍每种数学脚本函数。

函数	描述
Abs	返回指定数字的绝对值（无符号等价值）。
ArcCos	给定一个介于 -1 和 1（闭区间）的数字，返回一个介于 0 和 180 度的角度，其*余弦*等于该数字。
ArcSin	给定一个介于 -1 和 1（闭区间）的数字，返回一个介于 -90 和 90 度的焦点，且其*正弦*等于该数字。
ArcTan	给定一个数字，返回一个介于 -90 和 90 度的焦点，且其*正切*等于该数值。
Cos	返回给定角度（以度表示）的*余弦*。
Exp	返回 ***e*** 的某次幂。

函数	描述
Int	返回小于或等于指定数字的下一个整数。
Log	返回某个数字的自然对数值。
LogN	返回以 ***n*** 为底的 ***x*** 的*对数*值。
Pi	返回 *Pi* 的值。
Round	将实数舍入为指定的精度。
Sgn	确定值的符号 （正、零或负）。
Sin	返回给定角度 （以度表示）的*正弦*。
Sqrt	返回某个数字的*平方根*。
Tan	返回给定角度 （以度表示）的*正切*。
Trunc	通过简单地删除小数点右边的部分来截短一个实 （浮点）数。

如需有关每个函数的有效语法及其用法示例的详细信息，请参阅联机 *InTouch 参考指南*。

系统函数

系统函数用于在系统上执行动作，例如，激活另一个 Windows 应用程序，复制、删除或移动文件，以及检索应用程序的有关信息。系统函数有两种类型：“文件” 和 “信息”。“系统文件” 函数用于从文件读写数据。它们均有两个公共的参数：***Filename*** 和 ***FleOffset***。

Filename 指的是将从中进行读写的文件的名称。此文件名必须包含完整的路径。 ***FileOffset*** 指的是文件中开始进行读、写操作的位置 （从文件头开始，以字节计算）。文件的第一个字节为 ***FileOffset*** 0。执行完毕后，每个函数会返回文件刚刚读取或写入的数据之后的那个字节位置。例如，如果函数从字节位置 10 开始读取 5 个字节的数据，则该函数会返回 15。

FileOffset 标记名既可用作函数的参数，也可用作返回标记名。这样便可以实现连续操作。

示例：

FileOffset=FileReadMessage(*Filename,FileOffset,Message_Tagname*,*0*);

在上例中，从 ***Filename*** 读取一行文本。起始位置由 ***FileOffset*** 的初始值指定 （例如 0，即是指文件的开始位置）。随后，下一次读取的开始位置会被返回给 ***FileOffset***，以备下一次调用 **FileReadMessage( )** 时使用。每次调用函数时，随着 **FileReadMessage( )** 读取文件， ***FileOffset*** 会变得越来越大。

下面简要介绍每个系统脚本函数。

函数	描述
ActivateApp	激活当前正在运行的另一个 Windows 应用程序。
FileCopy	将 *SourceFile* 复制到 *DestFile*，它与 DOS 中的 Copy 命令或 Windows 文件中的 “复制”功能相似。
FileDelete	删除不必要或不需要的文件。
FileMove	类似于 FileCopy()，不同的是它将文件从一处移动到另一处，而不是复制一份。
FileReadFields	从指定的文件读取 “逗点分离变量”（Comma Separated Variable，简称 CSV）记录。
FileReadMessage	从指定的文件读取指定数量的字节 （或整行）。
FileWriteFields	将 “逗号分离变量”（Comma Separated Variable，简称 CSV）记录写入指定的文件。
FileWriteMessage	将指定数量的字节 （或整行）写入指定的文件。
InfoAppActive	测试应用程序是否在激活状态下。
InfoAppTitle	返回 “应用程序标题”或当前正在运行的特定程序的 Windows “任务列表”名。
InfoDisk	返回指定的本地 （或网络）磁盘驱动器的有关信息。
InfoFile	返回关于特定文件或子目录的信息。
InfoInTouchAppDir	返回当前的 InTouch 应用程序目录。
InfoResources	返回各种系统资源值，具体如下：**第 1 和第 2 种情形**：在 Windows NT 上， GDI 和 USER 采用硬编码方式，固定返回 50%。**第 3 种情形**：在 Windows NT 上，返回 “分页文件的可用字节”。**第 4 种情形**：在 Windows NT 上，返回搜索所有顶层窗口的结果。它仅计算可见的独立窗口。这不是系统中实际的 “当前运行的任务数”。与它最接近的数字是当您在 Windows NT 中运行 “任务管理器”时显示在 “应用程序”标签上的项目数。

函数	描述
IsAnyAsyncFunctionBusy	用于找出是否有任何异步 QuickFunction 正在运行。通过使用此函数，您可以让调用异步 QuickFunction 的 QuickScript 等待所有其它异步 QuickFunction 先完成处理。这样该 QuickScript 便能自我重新同步。
StartApp	自动启动另一个 Windows 应用程序。

如需有关每个函数的有效语法及其用法示例的详细信息，请参阅联机 *InTouch 参考指南*。

其它函数

其它一些函数可用于执行其它一些动作，如隐藏窗口、监控历史趋势、打印窗口、向其它应用程序发送键等。

下面简要介绍各种其它脚本函数。

如需新报警函数的列表，请参阅第 9 章 “报警 / 事件”。

如需有关新报警函数的详细信息，请参阅 *InTouch 参考指南*。

提示 以 "alm" 开头的函数名仅用于分布式报警系统。以 "wc" 开头的函数名用于 “窗口控件” 对象 （列表框、文本框、组合框等）。以 "HT" 开头的函数名仅用于历史趋势对象。

函数	描述
Ack	确认任何尚未确认的报警。此函数可应用于标记名、“报警组” 或 “组变量”。
almAckAll	确认当前查询中的所有报警，包括当前未显示在报警显示对象中的那些。
almAckDisplay	仅确认报警显示对象中当前可见的那些报警。
almAckRecent	确认最近发生的报警。
almAckSelect	仅确认报警显示对象中选定的那些报警。
almDefQuery	使用缺省属性执行查询以更新报警显示对象。
almMoveWindow	滚动报警显示对象窗口。
almQuery	执行查询以更新报警显示对象。

函数	描述
almSelectAll	切换报警显示对象中选择的所有报警。
almSelectItem	切换在报警显示对象中辉亮显示的选定项目。
almShowStats	显示报警显示对象统计屏幕。
ChangePassword	显示 “改变口令”对话框以便登录的操作员改变自己的口令。
DialogStringEntry	在屏幕上显示字母数字键盘，以便操作员在 “标记名字典”中改变消息型标记名的当前字符串值。
DialogValueEntry	在屏幕上显示数字小键盘，以便操作员在“标记名字典”中更改离散型、整型或实型标记名的当前值。
GetNodeName	将 NetDDE 节点名返回给一个字符串变量。
GetPropertyD	在运行时检索指定属性的离散值。
GetPropertyI	在运行时检索指定属性的整型值。
GetPropertyM	在运行时检索指定属性的消息值。
Hide	从脚本隐藏各种不同的窗口。 Hide 函数必须在每个要隐藏的窗口的名称前面。
HideSelf	隐藏当前的活动窗口。
HTGetLastError	确定上次检索指定的笔的过程中是否发生了错误。
HTGetPenName	返回指定趋势的指定笔号当前所用的标记名。
HTGetTimeAtScooter	返回样本自 1970 年 1 月 1 日 GMT 00:00:00 以来的时间 （以秒为单位），该样本位于 *ScootNum* 和 *ScootLoc* 指定的指示器位置。
HTGetTimeStringAtScooter	返回一个包含日期 / 时间的字符串，该日期 / 时间值针对 *ScootNum* 和 *ScootLoc*. 指定的指示器位置的样本。
HTGetValue	返回整个趋势的指定笔的请求类型值。
HTGetValueAtScooter	返回指定的指示器位置、趋势和笔号处的样本的请求类型值。

函数	描述
HTGetValueAtZone	返回某个趋势指定笔的左、右指示器位置之间的数据的请求类型值。
HTScrollLeft	将趋势的起始时间值设置为比当前起始时间早趋势宽度的一定百分比。其效果是将图表的日期 / 时间向左滚动给定的百分比。
HTScrollRight	将趋势的起始时间值设置为比当前起始时间晚趋势宽度的一定百分比。其效果是将图表的日期 / 时间向右滚动给定的百分比。
HTSelectTag	显示 "选择标记" 对话框，且操作员可以为指定的笔选择不同的标记名 （对话框仅列出 "标记名字典" 中已经给历史记录 （选定 "记录数据" 选项）定义的标记名）。
HTSetPenName	为趋势笔指定不同的标记名。
HTUpdateToCurrentTime	通过将结束时间设置为当前时间来检索并显示数据。开始时间将等于 EndTime （结束时间）减去图表的 "宽度"。
HTZoomIn	计算新图表的宽度和开始时间。如果趋势的 **.ScooterPosLeft** 是 0.0，且 **.ScooterPosRight** 是 1.0，则新图表的宽度将等于旧图表的宽度除以 2。
HTZoomOut	计算新图表的宽度和开始时间。新图表的宽度等于旧图表的宽度乘以 2。
IOReinitialize	关闭现有的所有 I/O 对话，并重新启动设置 I/O 对话的整个过程。执行此函数时，所有的 I/O 点均会受到影响。
IOGetApplication	将为特定 "访问名" 定义的应用程序名返回给指定的标记名。
IOGetNode	将为特定 "访问名" 定义的节点信息 （地址）返回给指定的标记名。
IOGetTopic	将为特定 "访问名" 定义的主题名返回给指定的标记名。

函数	描述
IOReinitialize	关闭现有的所有 I/O 对话，并重新启动设置 I/O 对话的整个过程。所有的 I/O 点均会受到影响。
IOSetAccessName	在运行时修改 “访问名”的*节点*、*应用程序*或*主题*名称部分，以便实施 InTouch 的热备份策略。
IOSetItem	改变 I/O 型标记名 ***.Reference*** 域中的访问名和 （或）项目。
LogMessage	将用户自定义的消息写入 Wonderware Logger。
PlaySound	通过 Windows 声音设备 （如已安装），播放 .wav 文件名或 WIN.INI 文件的 [sounds] 部分中的项目指定的波形声音。
PrintHT	可用在按钮中，以便打印与指定的 “历史趋势”型标记名相关的 “历史趋势”图表。使用此函数时，必须能够在屏幕上看到 “历史趋势”。
PrintScreen	打印指定的屏幕。
PrintWindow	打印指定的窗口。
ReloadWindowViewer	可供用户控制重新加载 WindowViewer。
RestartWindowViewer	可供用户控制 WindowViewer 的关闭和重新启动。
SendKeys	向应用程序发送键。
SetPropertyD	指定要在运行时写入的属性的离散值。
SetPropertyI	指定要在运行时写入的属性的整型值。
SetPropertyM	指定要在运行时写入的属性的消息值。
Show	显示指定的窗口。（窗口名必须用英文引号括起）。
ShowAt	指定窗口显示时的水平和垂直象素位置。窗口打开时，它将位于水平和垂直坐标的中心。

函数	描述
ShowHome	显示“起始”窗口。起始窗口是您选定的在 WindowViewer 启动时自动打开的窗口。（起始窗口是在 WindowViewer “属性 - 起始窗口”属性页中选定的）。 如需有关起始窗口的详细信息，请参阅第二章 - 使用 WindowMaker。
ShowTopLeftAt	指定窗口显示时它左上角的水平和垂直象素位置。窗口打开时，它的左上角将位于水平和垂直坐标的相交处。
wcAddItem	将提供的字符串添加到列表框或组合框中。
WcClear	从列表框或组合框删除所有的项目。
wcDeleteItem	删除与列表框或组合框中的项目索引参数关联的项目。
wcDeleteSelection	从列表中删除当前选定的项目。适用于列表框和组合框。
wcErrorMessage	给定的错误号， **wcErrorMessage()** 返回描述该错误的字符串消息。适用于列表框、文本框、组合框、单选钮和复选框。
wcFindItem	确定列表框或组合框中与提供的字符串相符的第一个项目所对应的索引。
wcGetItem	返回与列表框或组合框中相应的索引关联的项目字符串的值属性。
wcGetItemData	检索与列表框或组合框中的列表项关联的整型值。
wcInsertItem	将字符串插入列表框或组合框。
wcLoadlist	使用新项目替换列表框或组合框的内容。
wcLoadText	使用新字符串替换文本框的内容。
wcSavelist	使用列表对象中的项目替换文件名的内容。
wcSaveText	将文本框包含的文本保存到文件名。
wcSetItemData	将一个整数值指定给列表框中的某个项目。

如需有关每个函数的有效语法及其用法示例的详细信息，请参阅联机 *InTouch 参考指南*。

WW DDE 函数

您不应使用 WW DDE 函数来取代常规的 InTouch DDE 通讯。只要有可能，应尽量去创建 DDE 型标记名，从外部应用程序获取数据或向外部应用程序发送数据。 WWDDE 函数旨在支持那些不能使用 InTouch 支持的 DDE 进行通讯的应用程序。例如，有些应用程序仅支持 DDE “执行” 或 “插入”。

WWExecute()、**WWPoke()** 和 **WWRequest()** 函数使用与 MS Visual Basic (DDEML) 相同的 Windows 函数。一个简单的函数实际上可以做好几件事情。例如， **WWPoke()** 可以在执行 “DDE 初始化”、“DDE 插入” 和 “DDE 终止”，所有这些均在一个函数中完成。这使得 WWDDE 函数不易出错，但同时也降低了处理大量 DDE 消息的效率。作为使用这些函数的一般准则，切勿：

循环使用这些函数 （多次重复调用）。

在一行和相同的脚本中调用多个 DDE 函数。

使用它们在另一个 DDE 应用程序中调用冗长的任务。

如果 DDE 命令在另一个应用程序中执行冗长的任务，则它可能会用完所有可用的处理器时间。不过，即使在通讯发生困难时，也不会丢失任何数据。即使 “I/O 服务器” 不能给 InTouch 发送消息，它仍会继续尝试。

下面简要介绍每个脚本函数。如需有关每个函数的有效语法及其用法示例的详细信息，请参阅联机 *InTouch 参考指南*。

函数	描述
WWControl	可供从 InTouch “恢复”、“最小化”、“最大化” 或 “关闭” 应用程序。
WWExecute	将命令 （使用 DDE Execute）发送给指定的*应用程序*和*主题*。
WWPoke	将值插入 （使用 DDE Poke）指定的*应用程序*、*主题*和*项目*。
WWRequest	发出一个一次性请求 （使用 DDE Request）， 向特定的*应用程序*、*主题*和*项目*索取值。

脚本编辑器错误消息

如果脚本编辑器在验证脚本时检测到任何错误，它会显示一个相应的消息框。例如：



提示 在大多数情况下，遇到错误时， InTouch 会将光标放到脚本中出现错误的位置。不过，在有些情况下，例如缺少 ENDIF，光标会出现在脚本末尾。在系统会接受脚本之前，所有的错误必须均得以纠正。

错误消息	定义
仅可比较报警组是否相等	不能比较报警组的 <、 >、 <=、 >= 关系。
不能对此字符串执行加、减、乘、除等运算	字符串不支持这些运算。
不能把报警组与另一类型混合	尝试比较另一类型 （如整型）与 “报警组”，或尝试在需要使用 “报警组”的地方使用其它类型。
不要把字符串与另一类型混合	尝试将另一类型 （如整型）与字符串作比较，或尝试在需要使用字符串的地方使用其它类型。
无法取反报警组	已使用负号 (-)。
无法取反访问名	"-" 或 "~" 不允许用在 DDE “访问名”之前。
无法取反字符串	已使用负号 (-)。
无法取反 **TagID**	已使用负号 (-)。
无法取反窗口	"-" 或 "~" 不允许用在窗口名之前。
不能以此方式使用 “访问名”	在此环境下不能使用 DDE “访问名”。
不能以此方式使用 **HistTrendTag**	尝试将另一类型 （如整型）与字符串作比较，或尝试在需要使用字符串的地方使用其它类型。

错误消息	定义
不能以此方式使用 **TagID**	TagID 型变量不能在此环境下使用。
不能以此方式使用窗口名	窗口名不能在此环境下使用。
“E 格式**”**必须在 **-38** 和 **+38** 之间	最大的 "e" 格式数在 e-38 和 e+38 之间。
E 格式必须在 **E** 之后有数字	有效的 E 格式是 n.nnen，而 1.e 是无效的。
函数参数之后，要求有 **")"**	函数名后需要有一个匹配右括号以对应左括号。
函数名之后，要求有 **"("**	函数名后需要有一个左括号。
在 **0x** 后要求有一个数	在 InTouch 中可以输入十六进制（以 16 为基数）数。要输入十六进制数，请先写 0x 然后再跟数字。
要求 **IF** 之后是表达式	缺少离散型表达式。
函数需要模拟参数	此函数的参数要求使用一个模拟值。
需要另一个参数	函数所需的参数比目前提供的多。
需要另一个运算数	如果输入 "a + "，则 InTouch 会显示此错误消息。
要求赋值	在动作脚本中，输入了标记名，下一个逻辑运算应该会是赋值。
函数需要逗号及其它参数	此函数需要使用更多的参数。
需要 **DDE** 名称	在此环境下必须使用一个 DLL 名称。
需要 **ENDIF**	每个 IF 必须有一个对应的 ENDIF。
需要 **ENDIF** 或 **ELSE**	每个 IF/THEN 必须有一个 ENDIF 或 ELSE 与之对应。
等号 **(=)** 之后要求有表达式	在动作脚本中，输入了标记名和赋值符，但没有给赋值语句提供具体的值。如果输入的是 => 而不是 >=，也会导致此错误。
需要函数名	在此环境下必须使用 “函数名”。
需要右括号	未找到匹配的 ")"。
需要名称	此参数需要使用一个标记名。
语句中需要有名称	语句中缺少名称。

错误消息	定义
需要分号	行的结尾需要分号。
需要字符串	给定的参数必须是字符串表达式，例如，字符串标记名的名称或常量字符串 （用英文双引号 " 括起的文本）
需要 **THEN**	IF 语句后缺少 THEN。
需要窗口名 **-** 必须是字符串表达式	给定的参数必须是字符串表达式，例如，字符串标记名的名称或常量字符串 （用英文双引号 " 括起的文本）
额外表达式	例如，表达式 "a b" 无效，而 "a + b" 则有效。
函数仅在动作脚本或逻辑中有效	有些函数仅在脚本中有效，而在表达式中却不合法。
IF 表达式必须为离散型 （使用 == 勿使用 =）	因为使用赋值符 (=) 而不是比较符 (==) 导致出现此错误。例如， "IF a=b THEN …" 应该是 "IF a==b THEN…"。如果使用了 "IF x THEN" 且 x 不是离散型标记名，也可能会导致此错误。
运算数无效或缺少运算数	运算符所需的运算数无效或缺少。
占位符名无效 **-** 在 **?x** 后必须有字符	?x: 后面必须有描述性字符。
占位符名无效 **-** 第二个字符必须是 **d**、**i**、**a**、**r**、**m**、**v**、**g**、**h**、**t**	占位符名中描述的第二个数字字符无效。
占字符名无效 **-** 第三个字符必须是 **":"**	占位符名中描述的第三个数字字符无效。
逻辑 "与 / 或"必须使用离散型	AND/OR 运算符必须使用离散表达式。因此，如果 x 和 y 是离散型标记名，则 "x AND y" 正确；如果是任何其它类型，则会收到这个错误消息。
逻辑 "非"必须使用离散型	NOT 运算符必须使用离散表达式。因此，如果 x 是离散型标记名，则 "NOT x" 正确；如果 x 是任何其它类型，则会出现此错误消息。
最大字符串有 **131** 个字符	字符串超过所允许的最大长度。
必须指定函数的返回值	有些函数要求对返回值进行功能验证。
在小数点后面必须有数字	语法 "1." 无效。

错误消息	定义
此参数必须有历史趋势变量	在此环境下必须使用 “历史趋势” 型变量。
此参数点域必须使用可写的模拟变量或名称点域。	参数必须是整型或实型变量，或变量的整型或实型值。
此参数必须使用可写的离散变量	此函数的参数必须是离散型标记名，且 “未” 给它选定只读属性。
此参数必须使用可写的整型变量	此函数的参数必须是一个整型标记名，且 “未” 给它选定只读。
此参数必须使用可写的消息变量	此函数的参数必须是一个整型标记名，且 “未” 给它选定只读。
此参数必须使用可写的实型变量	此函数的参数必须是一个实型标记名，且 “未” 给它选定只读。
此参数必须使用可写的变量	此函数的参数必须是一个标记名，且 “未” 给它选定只读。
不能指定函数的返回值	有些函数不返回任何值，因此不能给它们计算返回值。
名称太长	标记名长度必须 <= 32。
没有右注释符	左注释分隔符 ({) 必须有与之匹配的右注释分隔符 (})。
没有右引号	缺少右引号 (")。
显示缓冲区中没有足够空间	没有足够内存用于此操作。释放内存，此操作便会完成。
表达式缓冲区中没有足够空间	此时没有足够内存用于此操作。释放内存，此操作便可完成。
0x 后只允许使用 **8** 位数字	输入十六进制数时，只允许使用 8 位数字。
数字太大	数字的绝对值必须 < 2e38。
数字太小	数字的绝对值必须 > 2e-38。
参数太多	提供的参数多于此函数所需的数目。
尝试赋值给一个只读的名称	给只读标记名赋值是无效的。

错误消息	定义
未定义域名	没有定义点域名，很可能是拼写错误。
无法识别的字符	辉亮显示的字符就表达式或动作脚本而言属无效字符。
使用了给标准标记名保留的域名（例如 **SP**）	标记名不能使用点域名。

窗口控件和分布式报警的错误消息

“窗口控件”和分布式报警 QuickScript 函数会根据 QuickScript 函数的处理结果来返回值。用于诊断错误的返回值可赋给 “整型”标记名。例如：

```
ErrorNumber = wcGetItem("ControlName", Number, Tagname);
```

其中：

ErrorNumber 是保存返回的错误值的**整型**标记名。函数的返回值可以传递给 **wcErrorMessage()**。 **wcErrorMessage()** 会返回错误的字符串描述。例如：

```
ErrorMsg = wcErrorMessge(ErrorNumber);
```

其中：

ErrorMsg 是保存返回错误的文本描述的**消息**型标记名。下表列出了这些数值及其描述：

错误消息	定义
0	成功
-1	一般错误
-2	可用内存不足
-3	属性为只读
-4	指定的项目已经存在
-5	未知的对象名
-6	未知的属性名
-x*	未知的错误

* *-x* 表示任何其它数字。

第 9 章

报警 / 事件

InTouch 提供了一个通知系统，可以向操作员通知生产过程与系统状况的有关信息。本系统支持过程报警与系统事件的显示、记录及打印。报警显示的介绍详见第 10 章 “报警 / 事件客户端”；报警记录器与报警打印程序的介绍详见第 11 章 “报警实用程序”；报警表示生产过程中发生的警报，而事件则表示正常的系统状态消息。

InTouch 支持 “分布式报警系统”，可供显示本地 InTouch 应用程序及其它网络 InTouch 应用程序的报警系统产生的报警与事件。这些报警可以在本地 InTouch 节点或从网络上的远程节点处确认。

本章介绍报警系统、各种不同类型的报警条件以及分组层次结构。具体的各小节介绍如何添加、修改及删除 “报警组”，如何给 “组”指定标记名、给标记名定义报警条件，以及如何配置报警系统。

目录

- 配置报警系统
- 给报警确认函数附加注释
- 确认本地报警
- 从旧的 InTouch 标准报警系统迁移到分布式报警系统
- 从旧的主 / 从报警系统迁移到分布式报警系统
- 热备份与同步
- 热备份用法示例
- 视图与存储过程列的定义
- 报警历史数据库视图
- 事件历史数据库视图
- 报警与事件历史数据库视图
- 报警计数器数据库存储过程
- 在 Enterprise Manager 中查看存储过程的定义
- 事件计数器数据库存储过程
- AlarmSuite 报警日志数据库视图

简介

InTouch “分布式报警系统”是独立于 WindowViewer 的一套软件组件。这些组件可供 InTouch 及其它 FactorySuite 程序在多节点系统中互相交流报警信息。充当*报警供应器*的 FactorySuite 程序执行报警状态的实际检测工作。“报警供应器”通过 API 调用将通知传递给 “分布式报警系统”。反之，充当*报警接收器*的 FactorySuite 程序则通过 API 调用从 “分布式报警系统”获取通知与状态，然后执行报警信息的显示工作。

在 InTouch 之前的版本中，“分布式报警系统”的重点放在通讯和一套 “报警接收器”上，这些报警接收器支持基本但却相当灵活的显示与存储功能：InTouch View 窗口中的报警显示、报警历史存储、报警记录以及报警打印等。增强的 “分布式报警系统”在改进性能的同时，还支持下列功能：

- 报警禁用与抑制
- 报警显示抑制
- 报警 SQL 数据库存储

“分布式报警系统”将继续提供显示、记录、打印以及确认生产过程报警及系统事件的服务。它处理本地 InTouch 应用程序及其它节点上运行的 InTouch 应用程序所产生的报警与事件。其它节点上的应用程序不必去识别。实际上，在配置为使用 “分布式报警系统”之后，甚至可处理非 InTouch 的 “报警供应器”的报警。

“分布式报警系统”的功能还包括：

- 显示与确认网络上任何 InTouch 节点上的报警的功能。

- ActiveX 报警显示，具有内置滚动条、可调大小的显示列、可选择多个报警、更新状态栏、动态显示类型以及基于报警优先级的显示颜色等功能。如需有关 ActiveX 报警显示的详细信息，请参阅第 10 章 “报警 / 事件客户端”
- QuickScript 函数，具备动态控制报警显示与报警确认的功能。如需有关 QuickScript 函数的详细信息，请参阅第 9 章 “报警 / 事件”
- 分组机制，具备让多个 “报警组”跨越不同的应用程序使用一个名称进行调用的功能。
- 确认时给报警添加注释的功能。

对于非 **InTouch** 报警供应器的支持

在过去， InTouch 一直是 FactorySuite 环境下的主 “报警供应器”（报警发生器）。 “分布式报警系统”从 InTouch 7.11 版开始支持非 InTouch “报警供应器”，例如， InBatch 与 InControl、“I/O 服务器”、报警打印服务器等其它一些 FactorySuite 组件，以及第三方组件等。

备注 要创建可通用与跟踪报警的第三方软件，则必须使用 Wonderware *报警工具包*。如需有关*报警工具包*的详细信息，请与您当地的 Wonderware 经销商联系。

分发应用程序

应用程序既可以手动分发，也可以使用 NAD 系统分发。分发应用程序时，报警组列表文件也会自动分发，如同它是应用程序的一部分。

如需有关 NAD 的详细信息，请参阅第 5 章 “建立分布式应用程序”

报警背景知识

下面是有关报警的一些常见术语和概念，而无论它们用于何种特定的系统、采用何种实现方式。

- **报警**：一般而言，**报警**是概念更为广泛的**条件**的一种特定类型 - 具体说来，报警是一种*异常条件*。通常，报警用于在发生某种错误或达到某个特定处理阶段时发出信号。例如，报警可能会指出锅炉超出安全温度极限，也可能只是简单地通知换班时间已到。
- **优先级**：**严重程度** （又称**优先级**）与报警关联，用于指明情况有多 “糟”，或者条件有多 “重要”。在锅炉超出温度极限时，严重程度可能 “非常糟”或 “非常重要”，需立即引起注意以确保生命与财产安全。与之相反，到达换班时间时，严重程度通常 “不至于太坏”，甚至非常轻微。报警的严重程度通常取决于具体环境 - 工厂应用、设备性质、后备系统的可用性、潜在损坏或停机成本等。

  备注 InTouch 使用一种**优先级标准**，其中 1 最重要， 999 最不重要。

  如需有关优先级的详细信息，请参阅 “报警优先级”。

- **子状态**：报警状态可能还包含**子状态**，在这种情况下它被称为**多状态报警**。例如，模拟报警通常有多个极限值，例如用 "High" 与 "Low" 用于限定正常操作范围，而用 "HiHi" 与 "LoLo" 表示极度偏离正常操作范围。上述锅炉温度水平就可以是在这些*子状态*之一的*报警*条件下。在继续处于整体*报警*条件时，也可以在任意两个子状态之间转换。

- **事件**：**事件**是*可以检测的实例*，它不一定与报警关联。进入或脱离某种报警状态这样的*转换*过程就一定会形成某种事件。事件也可以表示操作人员的动作、系统配置的改变或某些类型的系统错误。区分**条件**与**事件**这两个概念非常重要。**条件**可能持续几分钟、几小时、几天或几周。而**事件**则是瞬时的，在发生之后便立即结束。*报警*是一种条件，而*报警通知*则是一个事件。

- **确认**：报警的一个主要用途是向人员或系统的另一部分发出有关该条件的警告。该人员或系统应随之***确认***报警，表示已获悉该项报警。这与采取修正操作的问题没有关系，修正操作可能不会立即发生。它也同报警条件是否回到正常状态的问题没有关系，有时级别没有任何外界干预，它也可能恢复正常。确认只是表示有人已注意到报警的发生。高或中优先级的报警通常要求予以确认，而优先级非常低的报警则可能不要求确认。尽管产生报警的条件可能已消失（例如，过高的温度又降了下来），但在被确认之前，报警本身并不会被视为 “已经处理”。

  如需有关确认的详细信息，请参阅 “报警确认模型”。

- **组**：报警可能会被划分为各个**组**，以便于跟踪和管理。这些组可能表示工厂的不同区域、设备的各个部件、操作员的责任或者是特定的功能。组还可以按父组与子组这样的**层次结构**进行组织管理。

  如需有关组的详细信息，请参阅 “报警组”。

- **区域**：通常，工厂会被划分为各个**区域**，用以表示实际的位置、操作员的责任、工艺过程的各个阶段以及设备类型等等。与报警组类似，区域也可以通过划分为主区域和子区域来层次结构化。不过，考虑到一些工厂会将这两个概念区别对待，所以**区域**应与**组**分开使用。“分布式报警系统”不明确支持 “区域”的概念。不过，“报警组”可用于将报警划分成与 “区域”相对应的各个集合。

报警与事件

InTouch 使用两种通知方式来提醒操作员生产过程的活动情况：“报警” 与 “事件”。报警表示关于生产过程的警告，它要求操作员作出响应。通常，在过程值超过用户定义的极限时，例如模拟量超过上限值，便会触发报警。这会触发一种*未确认的*报警状态，它可以用于向操作员通知有问题发生。操作员确认报警之后，系统便会返回*已确认*状态。InTouch 可配置成要求即便引起报警的条件已消失但仍需确认报警。这可以确保操作员了解那些引起临时报警状态但已恢复正常的事件。

事件表示正常的系统状态消息，不要求操作人员作出响应。通常，发生特定的系统条件时，如操作员登录到 InTouch，则会触发一个事件。如果加以适当配置， InTouch 可以将事件记录到报警数据库，并 / 或在打印机上进行打印输出。

在 “标记名字典” 中定义标记名时，您可以配置任意标记名来执行事件监控。定义标记名来监控事件时，只要每次标记名值发生改变，均会有一则事件消息记录到报警系统。事件消息记录数据如何改变：是由操作员、 I/O、脚本还是由系统引起的改变。

如需有关设置标记名执行事件监控的详细信息，请参阅第 6 章 “标记名字典”

报警摘要与报警历史

InTouch 使用*报警摘要*与*报警历史*这两个词来分别指 “当前活动的” 报警和 “已经结束的” 报警。此理念基于，操作员可能希望查看目前等待确认的所有报警的 “摘要”，而所有其它报警信息已属于过去的历史，并非十分紧要。在 “分布式报警系统” 中，这两种**报警记录**分别保存在不同的存储缓冲区中。

终端服务报警支持

通过在 InTouch “终端服务” 中使用 “分布式报警系统”，运行在不同终端会话上的报警客户端可以选择要显示的哪些报警数据以及显示方式。

“报警供应器” 通过一个唯一识别其应用程序及其应用程序实例的名称来识别它们。“报警供应器” 或 “报警接收器” 向 “分布式报警系统” 注册之后，此信息将可供 “分布式报警系统” 使用。

“报警供应器” 所在节点则通过一个唯一识别系统中该计算机节点的名称进行识别。在计算机节点上启动 “分布式报警系统” 的一个实例之后，此信息便会提供给它。

在记录报警事件时，节点与完整的 “报警供应器” 名可确定该报警的来源。

备注 “终端” 会话尚不支持 “报警供应器”。它们只在 “终端控制台” 上受支持。

如需有关 “终端服务” 的详细信息，请参阅联机 *InTouch 终端服务布署指南*。

对其它报警源的支持

除 InTouch 外，“分布式报警系统” 还支持来自其它各种来源 （也称为 “报警供应器”）的报警。特别是在结合 Wonderware “I/O 服务器” 使用时，它还支持来自 RTU 的报警。在按批次模式接收 RTU 报警记录时，“分布式报警系统” 可以依序 “成批” 插入它们。

这些报警记录的日期 / 时间标签由 “报警供应器” 提供，而不是由 “分布式报警系统” 自动生成。

客户端发出的报警请求（譬如确认）由“分布式报警系统”发送消息给“报警供应器”来完成。“报警供应器”随后对消息进行内部处理，并回复最新的报警记录，指明出现的新状态或配置（如果有）。

“报警供应器”会将其特有的全部信息写入报警记录。“报警供应器”（而非“分布式报警系统”）会在报警记录中写入报警名、报警类、极限、启用状态、状态以及任何其它配置。

如需有关“终端服务”的详细信息，请参阅联机 *InTouch 终端服务布署指南*。

报警类型

InTouch 按照报警的特性将它们分为几种常见的类别。这些类别也就是所谓的*类*与*类型*。“分布式报警系统”按五种基本*条件*给所有的报警归类：离散、偏差、变化率、值以及 SPC。下表简要概括两种系统的报警条件：

报警条件	分布的类	分布的类型
离散	DSC	DSC
偏差 - 主	DEV	MAJDEV
偏差 - 副	DEV	MINDEV
变化率	ROC	ROC
SPC	SPC	SPC
值 - LoLo	VALUE	LOLO
值 - Low	VALUE	LO
值 - High	VALUE	HI
值 - HiHi	VALUE	HIHI

您可以将每种报警与 InTouch 标记名相关联。根据标记名的类型，您可以给它定义一个或多个报警类或类型。您可以在“标记名字典”中定义各种报警条件。

如需有关定义报警条件的详细信息，请参阅第 6 章“标记名字典”

您还可以配置报警打印和 / 或分布式报警显示对象或 Alarm Viewer ActiveX 控件，以显示报警*类*字段和 / 或报警*类型*字段。

事件类型

同样，InTouch 也按照事件的特性将它们划分为几个常见的类别。这些类别也就是所谓的*事件类型*。下表简要概括这些事件的分类情况：

事件	条件
SYS	发生了系统事件
USER	$Operator 发生改变
DDE	从 DDE 客户端插入了标记名值

事件	条件
LGC	QuickScript 修改了标记名值
OPR	操作员使用 “输入值”修改了标记名值

上面的前两个事件会在启用事件记录功能时自动配置。其余三个必须由用户针对 “标记名字典”中的每个标记名进行定义。

如需有关事件的详细信息，请参阅 “报警与事件”。

报警优先级

InTouch 中配置的每个报警都有一个关联的优先级值。此值表示报警的严重程度，范围从 1 到 999，其中 1 表示最为严重。使用这些优先级创建报警范围，将报警指定给每个优先级，用户可以很轻松地区分非关键报警与关键报警。您还可以根据优先级范围创建动画链接、确认脚本、过滤视图及打印。

例如，如果某个加工厂已确定他们需要使用四种不同的严重程度，则他们可以建立如下所示的各个范围：

报警严重程度	优先级范围
关键	1 - 249
主要	250 - 499
次要	500 - 749
提示性	750 - 999

工厂的工程师在创建 InTouch 标记名和报警条件时，通过选择上述范围内的优先级，为每个报警指定一个严重程度。配置好这些范围之后，现在工厂的操作员就可以轻松只显示与打印某种特定严重程度的报警。

备注 只使用四种优先级是可能的，甚至也建议您这样做：1 表示关键、2 表示主要、 3 表示次要、 4 表示提示性 （即 1、 2、 3、 4）。

报警组

每个 InTouch 报警均可指定给一个 “逻辑报警组”。这些组可由用户自定义，并可以整理成多达 32 级的层次结构。这些组提供了一种基于机构、工厂设计或任何其它衡量标准来给报警归类的方式。对于过滤报警显示、“报警打印程序”及确认脚本，“报警组”均非常有用。

每个标记名都与一个 “报警组”关联。如果没有将 “报警组”名与标记名关联，则缺省条件下， InTouch 会自动将它与根报警组 **$System** 关联。任何 “报警组”均可以有与之关联的标记名及其它 “报警组”名。“报警组”组织整理为分层的目录树结构，根报警组 **$System** 位于目录树的顶端。定义的所有 “报警组”均自动成为根报警组的下级。

此目录树最多可达 32 级。每个 “报警组” 最多可以有 32 个子组。每个子组最多可以有 32 个子组，直至级数达到最大值 32 为止。



上图仅显示 “报警组”，并不显示每个组内的标记名。此目录树的概念类似于 Windows 的文件夹目录结构，其中，一个目录可以包含其它子目录 （类似于组）与文件名 （类似于标记名）。

“分布式报警系统” 还将这些组用作其 “报警组列表” 的基础。

如需有关详细信息，请参阅 “分布式报警组列表”。

备注 尽管在 InTouch 授权方式中 “报警组” 并不算作标记名，但在数据库中它们却会被视为标记名。因此，“报警组” 总数加实际的标记名数量不能超过 61,402。

警告！ 报警组数过大 (100/1000) 会导致性能问题。

要创建报警组

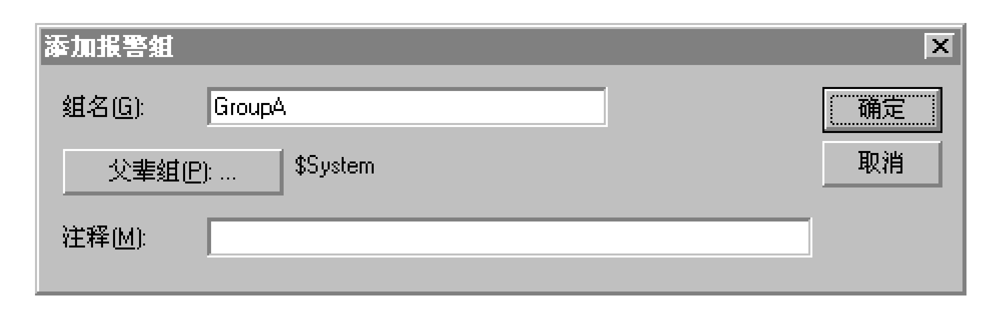
1. 在**特别**菜单上，选择**报警组**。此时会出现**报警组**对话框。

   提示 您也可以创建 “报警组”，并在 “标记名字典” 中定义标记名时将它与标记名关联。



2. 单击**添加**。此时会出现**添加报警组**对话框。

提示 在定义“报警组”之后，**修改**与**删除**按钮才可用。**$System** “报警组”不得修改或删除。

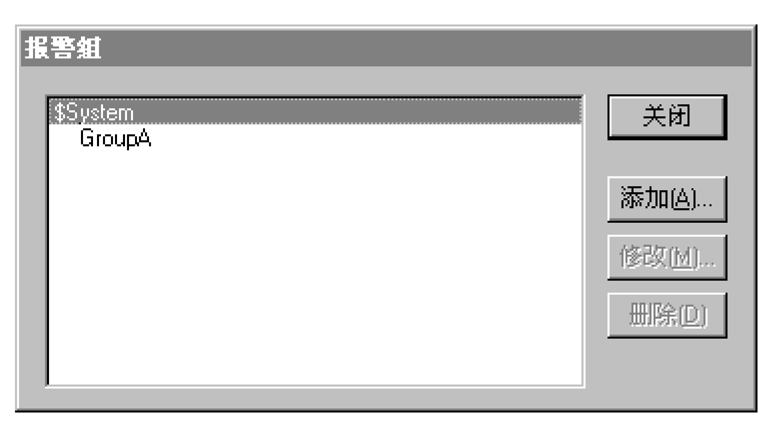


提示 如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

3. 在**组名**框中，输入新“报警组”的名称。因为这是您创建的第一个“报警组”，则它会自动分配到 **$System** “父组”。

提示 在创建一个“报警组”之后，便可以将它用作“父组”。

4. 单击**父组**以便将您的“报警组”指定给一个不同的“父组”。此时会出现**报警组选择**对话框。

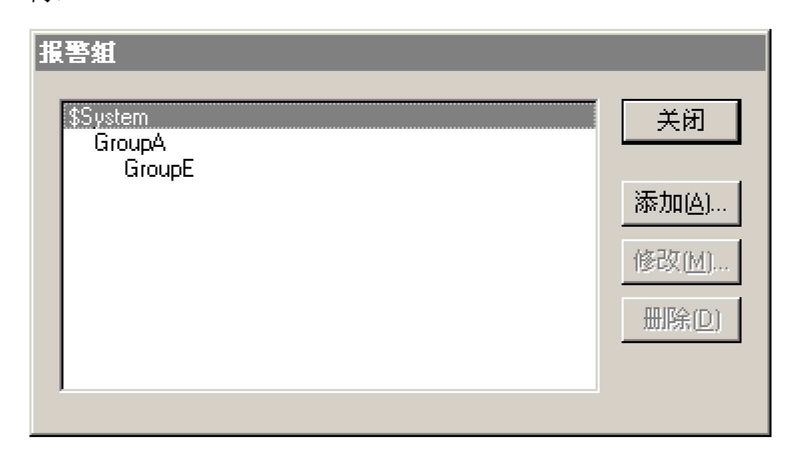


5. 在**选择报警组**列表中，双击要用作新“报警组”的“父组”的“报警组”名称（或选择它，然后单击**确定**）。此时会再次出现**添加报警组**对话框，显示所选的“父组”。例如：



6. 在**注释**框中，给新“报警组”输入任意注释。

7. 单击**确定**。此时会出现**报警组**对话框，显示您的 “报警组”层次结构：

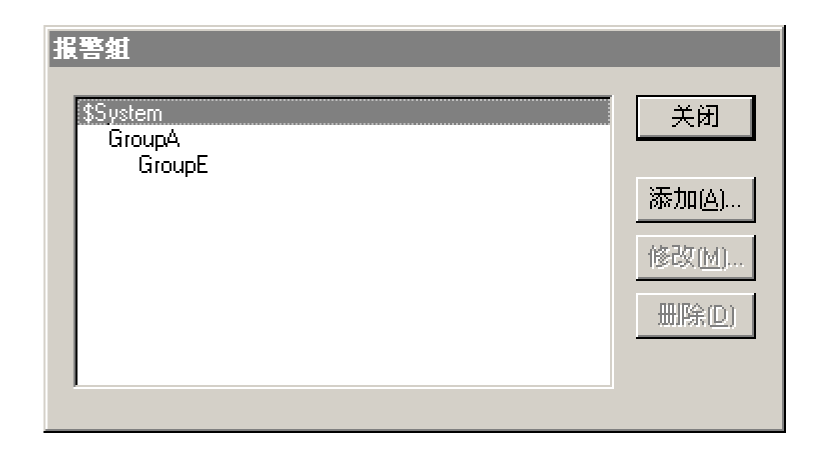


8. 单击**关闭**。

要修改报警组

1. 在**特别**菜单上，选择**报警组**。此时会出现**报警组**对话框。

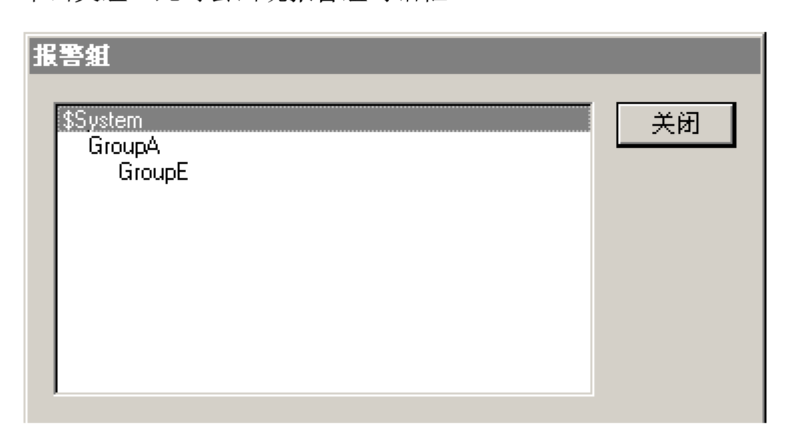
提示 您还可以在 “标记名字典”中定义标记名时修改 “报警组”。



2. 在列表中选择要修改的 “报警组”，然后单击**修改**。此时会出现**修改报警组**对话框。

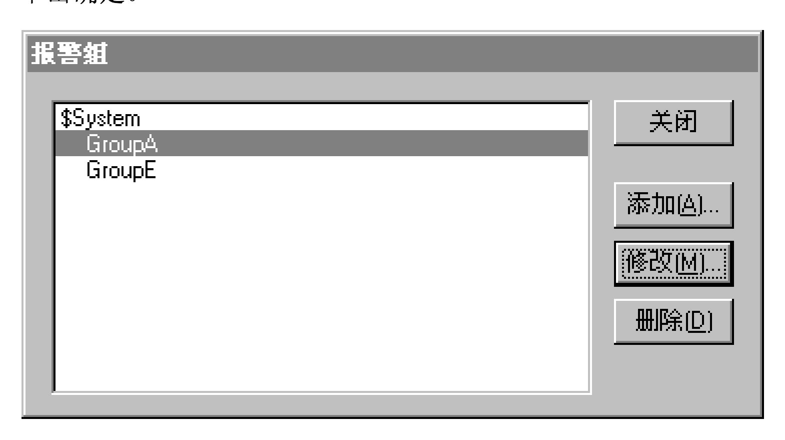


对 “报警组”进行所需的更改。如果要更改 “报警组”的父组，请单击**父组**。此时会出现**报警组**对话框。



选择新的父组，然后单击 “关闭”。此时会出现 “修改报警组”对话框，显示新的父组。

3. 单击**确定**。



要删除报警组

1. 在**特别**菜单上，选择**报警组**。此时会出现**报警组**对话框。

提示 您还可以在 “标记名字典”中定义标记名时删除 “报警组”。

报警组

$System
GroupA
GroupE

关闭
添加(A)...
修改(M)...
删除(D)

2. 在列表中选择要删除的 “报警组”，然后单击**删除**。此时会出现一个消息框，要求您确认删除。单击**是**以删除 “报警组”，或单击**否**以取消删除。
3. 单击**关闭**。

备注 为已删除的组配置的任何报警均会自动指定给父组。

分布式报警组列表

“分布式报警系统”使用 “报警组”机制将报警归类到本地分层目录树结构，供分布式报警显示用来过滤要显示的报警。“分布式报警系统”还可供您查看网络上多个节点的报警组。 此外，“分布式报警系统”还可供您通过创建**报警组列表**来整理报警视图。如需有关查看报警的详细信息，请参阅第 10 章 “报警 / 事件客户端”

报警组列表是一个有名称的列表，它由 InTouch 节点与每个这些节点上定义的 “报警组”组成。它还可以包含其它 “报警组列表名”及本地 “报警组”。“报警接收器”（如 “分布式报警显示”）使用此列表来查询报警。如需有关 “分布式报警显示”的详细信息，请参阅第 9 章 “报警 / 事件”

例如，如果对跨越多个 InTouch 节点的所有锅炉报警感兴趣，则可以建立一个名称为 "BoilerAlarms" 的查询。附加到该查询的列表包含对应于锅炉报警的所有节点上的所有 “报警组”。

要创建报警组列表

1. 在**特别**菜单上，指向**配置**，然后单击**分布式名称管理器**。此时会出现**分布式名称管理器**对话框，并显示**分布式报警**属性页：

提示 要快速访问该对话框，请在 “应用程序浏览器”中的**配置**下，双击**分布式名称管理器**。



2. 在**组属性**区，在**名称**框中输入查询的名称。
3. 在**成员**框中，输入要纳入查询的 InTouch 节点与 “报警组”的列表。这些列表的有效语法包括：

   标准组项

\\Node\InTouch!Group	远程节点上 “报警组”的全限定路径
\InTouch!Group	同上，但假定 “节点”为本地
GroupList	另一个 “组列表”

   快捷组项：

Node.Group	等价于 **\\Node\InTouch!Group** 的快捷方式
.Group	等价于 **\InTouch!Group** 的快捷方式

   其中， **Node** 表示 InTouch 远程节点的名称，而 .Group 表示该节点上的 “报警组”。如果 “报警组”在本地，则仅可输入带句点的 “报警组”名。例如 **.AlarmGroup**。

提示 快捷组项提供了一个简便的方法，可以很容易地将节点与 “报警组”信息输入对话框。保存 “报警组列表”时，切记将此信息转换成**标准组项**格式，这点非常重要。

备注 **Node.Group** 与 **.Group** 语法仅可用在此配置对话框中。在报警显示配置或任何报警 QuickScript 函数中，它是无效的。

4. 单击**添加**将此列表添加到 “报警组”文件。此时会自动转换**成员**的语法。

5. 单击**确定**。

报警确认模型

本节介绍如何报告、跟踪及确认报警。

只要某个项目从**正常**状态转到**报警**状态时，便会产生其报警的一个新*实例*。如果该报警属多状态报警，则仍保持报警状态时改变到新的子状态均会被视为相同报警实例的转换。在报警返回**正常**状态时，报警实例的*生命周期*便会结束 - 以后再转换为报警状态时，会生成一个新的报警实例。

InTouch “分布式报警”系统会跟踪报警的每个*实例*，无论是首次进入报警状态、进行子状态转换、恢复**正常**状态、等待**确认**，还是获得**确认**时。

- **条件报警**：确认是针对报警的*实例*进行的。报警实例在首次进入**报警**状态时，它开始等待**确认**。如果获得**确认**，并随后转入新的报警子状态 （例如从 "Hi" 转入 "HiHi"），则它开始等待下一次**确认**。只要得到**确认**，它均会被接受并应用于迄今为止出现的报警的所有状态转换。最新的实例得到**确认**时，报警即被视为**已确认**。

- **事件报警**：确认是针对报警的*实例*进行的，并且必须针对进入报警状态或子状态的最近一次转换。在首次进入报警状态时，报警*实例*开始等待**确认**。如果得到**确认**，并随后转入新的报警子状态，则它开始等待下一次**确认**。每次后续转换均会被指定一个序号，**确认**必须在其后附上与其对应的转换的序号。**确认**只有在针对最近一次转换时才会被接受。被拒绝的**确认**可能会被记录作诊断用途，但不会以其它方式在系统中跟踪。如果被接受，则它将应用于迄今为止出现的报警的所有状态转换。最新的实例得到**确认**时，报警即被视为**已确认**。面向事件的模型可确保报警在不同状态间变化时，**确认**仍可对应最新的信息。在等待时间较短的系统中，这可能看上去有些象是面向条件的报警 - 但在如 Internet 这样的其它环境，此模型的功能便显得非常重要。

- **扩展的报警摘要**：确认是针对报警的每次转换进行的。首次进入报警状态必须得到**确认**，恢复为正常状态也必须单独进行**确认**。到新报警子状态的任何转换都被视为必须**确认**的新事件，其 **RTN** 也必须**确认**。子状态转换会被视为属于 "**RTN 组** "，自项目从先前的*正常*状态首次进入*报警*状态开始。如果该项目*恢复正常*，随后再次进入*报警*状态，则会创建新的 **RTN 组**。每次转换均必须明确地单独加以确认；只有在项目*恢复正常*并且所有待定 **RTN 组**中的所有转换均得到确认时，报警才被视为***已确认***。

备注 此处使用 “摘要”一词来表示 “等待确认”的意思。此报警模型在工业中也称为 “回铃报警”。

概括地说，在**面向条件的**报警中，确认取决于进行确认时进入*报警*状态的所有项目。对于**面向事件的**报警 （如在 OPC 中），只有指最近一次“激活”事件的情况下，确认才会被接受。对于**扩展的摘要**报警，所有事件 - 进入 “活动”状态、“不活动”以及不同的子状态 - 均必须得到确认，整个报警才被视为已确认。

扩展的报警摘要

发生报警时，它会在报警显示对象中生成一条记录，显示已发生的报警条件及报警的日期与时间标签。此记录将一致保留在显示对象中，直到操作员确认报警并发生恢复正常状态 (RTN) 事件。如果在确认报警之前发生恢复正常状态事件，则报警对象中会显示两条记录。

例如，如果锅炉温度超出上限状态并因此触发报警，但在操作员确认报警状态之前它已恢复正常温度范围，则会生成一条记录显示报警状态，同时生成另一条记录，显示报警状态尚未得到确认。

确认只是针对特定的状态转换，无论是进入报警状态、子状态还是恢复正常状态。每次从正常状态的转换均标志着一个新 RTN 组的开始。RTN 组中所有的转换均必须单独予以确认，之后整个 RTN 组才被视为已确认。

使用扩展的报警摘要

在定义标记名并选择**扩展的摘要**作为其确认模型时，要求操作员确认发生的报警，即便触发报警的状态现已恢复正常。确认报警状态时，会改变报警项的颜色，但不会改变所显示的时间标签。只有在报警得到确认且报警已恢复正常状态时，报警才会从显示对象中清除。

备注 使用**扩展的摘要**确认模型定义标记名时，不得将**报警属性**对话框中的**返回意味着确认**选项应用于该标记名。

如需有关详细信息，请参阅第 6 章 “标记名字典”

发布 / 预订机制

在任何给定的节点上，均会有**报警供应器** （发布者）与**报警接收器** （预订者）集合。InTouch “分布式报警系统”提供**发布 / 预订机制**在节点之间和软件组件之间传递报警信息。

报警供应器

报警供应器会跟踪*应报警*项 - 即可能进入*报警*状态的项目 - 并向 “分布式报警系统”提供这些项目的列表，包括有关项目组的任何层次结构的信息。它也会在项目状态改变时通知 “分布式报警系统”。状态改变包括项目是进入还是脱离*报警*状态，以及是否*已确认*最近的报警。

此外，“报警供应器”还会跟踪项目是否被禁用。

备注 在 Windows 2000 中，如果 “报警供应器” 与 “报警接收器” 在不同的域中，则 “报警接收器” 将无法查看报警，除非查询使用供应器设备的全限定名称或供应器名称的 IP 地址。例如，可以在查询中将另一个域中的 “报警供应器” 指定为：
\\provider1.b3.wonderware.com\intouch!$system
其中， "provider1" 是机器名， "b3.wonderware.com" 是域的 “主 DNS 后缀”。

报警接收器

报警接收器给 “分布式报警系统” 提供一套**查询**，以确定它想要收到的报警项目的通知。在 “报警接收器” 更改或删除它之前，**查询**将保持活动状态，并指定 “报警供应器” 或报警组 - 这与使用 “通用符” 的 SQL 查询非常相似。只要 “报警供应器” 发出更改通知时，“分布式报警系统” 便会检查报警是否与已注册的任何查询匹配，然后将更新传递给相应的 “报警接收器”。在收到更新之后，“报警接收器” 即显示或记录与项目的状态或其转换有关的信息。报警 “接收器” 也可***确认***报警。即发送确认通知给 “分布式报警系统”，确定报警与 “报警供应器”。该通知会被传递给 “报警供应器”，后者随之将项目状态更新为***已确认***（如果适当），并进而通知 “分布式报警系统”，从而确保将更新发布给所有相关的 “报警接收器”。

备注 “分布式报警系统” 中的大部分通讯活动是将**报警查询**与**报警记录**从一个节点发送到另一个节点。在同一个节点中，报警查询与报警记录由**报警缓冲区**记录并缓存，以最大限度减少网络流量。

报警数据存储

“分布式报警系统” 使用多种数据存储形式：

- **报警缓存：**有关当前与最新报警的大部份信息均缓存于各计算机节点的内存中。最初的 InTouch 报警系统使用两个高速缓冲存储器：一个用于存储 “摘要”（当前）报警，一个用于存储 “历史” 报警。此模型也在 “分布式报警系统” 中使用。

  摘要 （当前）报警的缓存可根据需要扩展以容纳所有的当前报警，直至达到可用内存的极限。报警历史的缓存只能扩展到 .INI 文件中配置的预设极限。历史缓存达到此极限之后，随着新的报警记录加进来，旧的报警记录会被丢弃。在多节点环境下，各个节点上的报警缓存共同组成一个分布式内存数据库。

- **报警记录：**Alarm DB Logger 会创建一个数据库，以记录报警发生、子状态转换、确认及恢复正常的时间。实际上，这些数据会形成系统中的永久或准永久性报警历史记录。

“分布式报警系统” 支持将报警记录到 Microsoft SQL Server 或 MSDE 之类的数据库。这些数据库支持通过开放的非专有接口进行访问，因而可以轻松检查或分析数据库内容。

备注 最初的 InTouch 标准报警系统使用可由用户配置的格式将报警记录到一个**纯文本文件**。此种形式的记录功能已经随同 “标准报警子系统” 一起退出历史舞台。

“分布式报警系统” 以**查询**为基础，因而支持使用一个计算机节点来记录多个其它节点的报警。

标记名报警配置

InTouch 允许您为在 “标记名字典” 中定义的每个标记名定义报警配置。（缺省条件下，所有标记名均禁用其报警）。基本思想是通过指定报警类型和用户自选极限，任何标记名都可以配置为*应报警*。对于这样的标记名，只要标记名的值发生改变，**报警逻辑**均会被激活。此报警逻辑是 InTouch 的内部子程序，它通过检查报警类型、比较新值与给定的极限，来确定标记名是否在*报警*状态下。随后任何状态转换均会被报告给 “分布式报警系统”。

InTouch 中定义了三种基本报警类型 （或类）。这些又可细分为七种附加的报警子类型：

- **离散**：离散报警对应于离散型标记名。您可以配置*报警*状态是对应于离散型标记名的 TRUE 状态还是 FALSE 状态，以及相关的报警*优先级*。离散型标记名的配置对话框如下：



模拟：模拟报警对应于整型或实型 （浮点）标记名。模拟类型共有七个报警类：

- **值**：当前值会与一个或多个极限比较。如果值超过极限，则表示出现报警状态。您可以分别配置 "LoLo"、"Lo"、"Hi" 及 "HiHi"' 等极限的值与优先级，并指明是否使用每个极限。
- **偏差**：当前值会与目标值比较，差的绝对值会与一个或多个极限比较，这些极限用标记名值*范围*的百分比表示 - 即配置的最大允许值与最小允许值之间的总差值。您可以分别配置 “副偏差” 极限和 “主偏差” 极限的值与优先级，并指明是否使用每个极限。您也可以配置 “偏差死区” 值，它也是用标记名范围的百分比表示。此值控制标记名值的百分比必须位于何种范围内才不致引发报警。

- **变化率**：此计算方式中会使用当前值和先前值，以及当前时间与上次更新时间。如果变化率的绝对值超出极限，则表示出现报警状态。您可以配置 “变化率” 极限的值与优先级，以及是否使用该极限。此极限用标记名值范围 - 可以是秒、分钟或小时 - 的百分比表示。模拟型标记名的配置对话框如下：



SPC：SPCPro 程序可以生成 “统计过程控制” 报警。这实际是在 InTouch 之外定义的，并通过与报警的其它部分不同的软件执行途径操作。

如需有关 SPC 报警的详细信息，请参阅联机 *SPCPro 用户指南*。

如需有关定义报警条件的详细信息，请参阅 “定义标记名报警条件”。

报警点域

InTouch 提供了各种报警 **“点域”**，供您动态控制和 （或）监视各种报警条件。许多这样的**点域**可通过使用 I/O、表达式和 （或）脚本进行访问。I/O 访问提供了利用其它 Windows 应用程序（如 Excel 或远程 View 应用程序，这会在本章稍后部分介绍）来监视和 （或）控制特定标记名的报警信息的功能。

例如，如果您创建一个名称为 **Analog_Tagname** 的模拟报警标记名，它将具有相关的一些 “属性”，如名称、 **HiHi** 设定点等等。这些 “属性” 中的一些可通过逻辑脚本、表达式及用户输入访问，它们也被称为**点域**。

访问与标记名关联的点域要求使用的语法是 **Tagname.dotfield**。例如，如果您希望允许运行时更改 **Analog_Tagname** 上的 **HiHi** 报警限，则可以给某个按钮创建一个**模拟 - 用户输入**触动链接，并在该链接的对话框中输入 **Analog_Tagname**.**HiHiLimit** 作为表达式。在运行期间，操作员只要单击按钮，并为要用于 **Analog_Tagname** 的 **HiHi** 报警限输入一个新值即可。

下面简要介绍有关如何使用报警点域的几个示例。

点域	描述
.Ack	可读 / 写离散型标记名点域，用于监视 / 控制标记名与 “报警组” 的报警确认状态。 **.Ack** 有一个逆向标记名点域 **.Unack**。发生未确认的报警时， **.UnAck** 将设置为 1。 **.UnAck** 随后可用在动画链接或条件脚本中，以针对未确认的报警触发电铃。
.AckDev	监视 / 控制标记名上活动偏差型报警的报警确认状态。
.AckDsc	监视 / 控制离散型标记名的当前确认状态。
.AckROC	监视 / 控制标记名上活动变化率型报警的报警确认状态。
.AckValue	监视标记名上活动值类型报警的报警确认状态。
.Alarm	发出存在报警条件的信号。
.AlarmAccess	返回与所选报警关联的标记名的名称。报警必须通过单击 “分布式报警显示” 摘要来选取。
.AlarmAckModel	监视与标记名关联的**确认模型**，具体如下： 0 = 条件 （缺省值） 1 = 事件 2 = 扩展 适用于含报警的离散或模拟型标记名。只读，但可以在 WindowMaker 中进行配置。
.AlarmClass	返回当前报警的类。
.AlarmComment	可读 / 写文本字符串，描述报警而非标记名的内容。缺省条件下，它在新应用程序中为空。 不过，在将 InTouch 7.1 应用程序转换为 7.11 应用程序时，为保证后向兼容性，标记名注释会被复制到 **AlarmComment** 中。
.AlarmDate	返回当前报警的日期。
.AlarmDev	发出存在偏差报警的信号。
.AlarmDevCount	记录给定标记名或报警组上活动偏差报警的总数。
.AlarmDevDeadband	监视 / 控制副偏差与主偏差报警偏离死区的百分比。
.AlarmDevUnAckCount	记录给定标记名或报警组上活动的未确认偏差报警总数。

点域	描述
.AlarmDisabled	禁用或启用事件与报警。适用于含报警的离散或模拟型标记名，或是报警组。 **备注** 此点域与 **.AlarmEnabled** 相同，但极性相反。
.AlarmDsc	指出离散报警条件当前处于活动状态。
.AlarmDSCCount	记录给定标记名或报警组上活动离散报警的总数。
.AlarmDscDisabled	指出标记名是否可生成离散型报警。 **备注** 对于离散型标记名，此点域与 **.AlarmDisabled** 相同。
.AlarmDscEnabled	指出标记名是否可生成离散型报警。 **备注** 对于离散型标记名，此点域与 **.AlarmEnabled** 相同。
.AlarmDscInhibitor	返回指定给此标记名的离散报警 （如果有）的约束标记名。
.AlarmDscUnAckCount	记录给定标记名或报警组上活动的未确认离散报警总数。
.AlarmEnabled	禁用或启用事件与报警。
.AlarmGroup	包含用于填写分布式报警显示对象的当前查询。
.AlarmGroupSel	返回报警的报警组。
.AlarmHiDisabled	禁用或启用含报警的模拟型标记名的 **Hi** 报警限。
.AlarmHiEnabled	禁用或启用含报警的模拟型标记名的 **Hi** 报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmHiInhibitor	返回 **Hi** 报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmHiHiDisabled	禁用或启用含报警的模拟型标记名的 **HiHi** 报警限。
.AlarmHiHiEnabled	禁用或启用含报警的模拟型标记名的 **HiHi** 报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmHiHiInhibitor	返回 **HiHi** 报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmLimit	返回当前报警的报警限。

点域	描述
.AlarmLoDisabled	禁用或启用含报警的模拟型标记名的 **Lo** 报警限。
.AlarmLoEnabled	禁用或启用含报警的模拟型标记名的 **Lo** 报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmLoInhibitor	返回 **Lo** 报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmLoLoDisabled	禁用或启用含报警的模拟型标记名的 **LoLo** 报警限。
.AlarmLoLoEnabled	禁用或启用含报警的模拟型标记名的 **LoLo** 报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmLoLoInhibitor	返回 **LoLo** 报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmMajDevDisabled	禁用或启用含报警的模拟型标记名的**主偏差**报警限。
.AlarmMajDevEnabled	禁用或启用含报警的模拟型标记名的**主偏差**报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmMajDevInhibitor	返回**主偏差**报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmMinDevDisabled	禁用或启用含报警的模拟型标记名的**副偏差**报警限。
.AlarmMinDevEnabled	禁用或启用含报警的模拟型标记名的**副偏差**报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmMinDevInhibitor	返回**副偏差**报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmName	返回当前报警的名称。
.AlarmOprName	返回当前报警的操作员。
.AlarmOprNode	返回当前报警的操作员节点。
.AlarmPri	返回当前报警的优先级值 (1-999)。

点域	描述
.AlarmProv	返回当前报警的供应器。
.AlarmROC	发出存在变化率型报警的信号。
.AlarmROCCount	记录给定标记名或报警组上活动变化率报警的总数。
.AlarmROCDisabled	禁用 / 启用含报警的模拟标记名的**变化率**报警限。
.AlarmROCEnabled	禁用 / 启用含报警的模拟标记名的**变化率**报警限。 **备注** 与对应的禁用点域相同，但极性相反。
.AlarmROCInhibitor	返回**变化率**报警限的约束标记名引用。适用于含报警的模拟型标记名。 只读，但可以在 WindowMaker 中进行配置。
.AlarmROCUnAckCount	记录给定标记名或报警组上未确认变化率报警的总数。
.AlarmState	返回当前报警的状态。
.AlarmTime	返回当前报警的时间。
.AlarmTotalCount	记录给定标记名或报警组上活动报警的总数。
.AlarmType	返回当前报警的类型。
.AlarmUnAckCount	记录给定标记名或报警组上活动未确认报警的总数。
.AlarmUserDefNum1	可读 / 写实型 （浮点）值，缺省值为 0，且值未设置。适用于含报警的离散型标记名、含报警的模拟标记名，或是报警组。 **备注** 此点域的值会附加在报警上，但 “只”限于已设置 （例如通过脚本或插入）值的情况。
.AlarmUserDefNum2	可读 / 写实型 （浮点）值，缺省值为 0，且值未设置。适用于含报警的离散型标记名、含报警的模拟标记名，或是报警组。 **备注** 此点域的值会附加在报警上，但 “只”限于已设置 （例如通过脚本或插入）值的情况。
.AlarmUserDefStr	可读 / 写文本字符串，缺省值为 ""，且未设置值。适用于含报警的离散型标记名、含报警的模拟标记名，或是报警组。 **备注** 此点域的值会附加在报警上，但 “只”限于已设置 （例如通过脚本或插入）值的情况。
.AlarmUserDefNumSet1	可读 / 写离散型。脚本的 TRUE 情形已定义了对应标记的 .AlarmUserDefNum1。要取消定义该标记的 .AlarmUserDefNum1 的值，请将此点域设置为 FALSE。缺省值为 FALSE。

点域	描述
.AlarmUserDefNumSet2	可读 / 写离散型。脚本的 TRUE 情形已定义了对应标记的 .AlarmUserDefNum2。要取消定义该标记的 .AlarmUserDefNum2 的值，请将此点域设置为 FALSE。缺省值为 FALSE。
.AlarmUserDefStrSet	可读 / 写离散型。脚本的 TRUE 情形已定义了对应标记的 .AlarmUserDefStr。要取消定义该标记的 .AlarmUserDefStr 的值，请将此点域设置为 FALSE。缺省值为 FALSE。

备注 **.AlarmUserDefNum1**、 **.AlarmUserDefNum2** 以及 **AlarmUserDefStr** 点域可供您在报告时指定一个或多个要追加到报警记录的值。这些值由 Alarm DB Logger 写入数据库。您可以在报警上附加三个项目：两个数字和一个字符串。缺省条件下，它们为空 （零和 ""）。

要简化设置用户值，您可以在报警组及特定的标记名上设置它们。例如， InBatch 可以在 **$System** 报警组上设置 **.AlarmUserDefNum1** 中的批号，从而让所有的报警均附加上该批号。

如果在某个 “报警组” 上设置 **.AlarmUserDefNum1**，则它将应用于该组及其子组中的所有报警。您也可以在标记名上专门设置 **.AlarmUserDefNum1** 的值。在这种情况下，它只应用于该标记名，并且会改写标记名的 “报警组” 中 **.AlarmUserDefNum1** 的任何设置。您可以通过将相应的 “设置” 点域设置为 FALSE 来取消定义用户定义的点域。例如，您可以给 Tag 1 设置 .AlarmUserDefNum1Set = FALSE 来取消定义 Tag 1 的 .AlarmUserDefNum1 的 值。

点域	描述
.AlarmValDeadband	监视 / 控制报警死区的值。
.AlarmValue	返回当前报警的值。
.AlarmValueCount	记录给定标记名或报警组上活动值报警的总数。
.AlarmValueUnAckCount	记录给定标记名或报警组上活动未确认值报警的总数。
.DevTarget	监视 / 控制副偏差与主偏差报警的目标。
.Freeze	读取 / 写入 “分布式报警显示对象” 的冻结状态。
.HiLimit, .HiHiLimit, .LoLimit, .LoLoLimit	可读 / 写模拟型标记名点域，用于监视或控制值报警检查的极限值。这些点域仅对于整型和实型标记名才有效。
.HiStatus, .HiHiStatus, .LoStatus, .LoLoStatus	只读离散点域，确定是否存在指定类型的报警。这些点域仅对于整型和实型标记名才有效。

点域	描述
.MajorDevPct	可读 / 写整型点域，用于监视或控制报警检查的主偏差百分比。
.MajorDevStatus	只读离散型点域，确定是否存在指定标记名的主偏差报警。
.MinorDevPct	可读 / 写整型点域，用于监视和 （或）控制报警检查的副偏差百分比。
.MinorDevStatus	只读离散型点域，确定是否存在指定标记名的副偏差报警。
.Name	可读 / 写消息型点域，用于显示标记名的实际名称。例如，它可以用于确定 “组变量”所指向的 “报警组”的名称或 TagID 标记名的名称。它也可以被改写，以更改 “组变量”所指向的 “报警组”。
.ListChanged	指出是否存在任何新报警或 “分布式报警对象”的更新。
.Normal	只读离散型点域，指定的标记名不存在报警时它等于 1。此点域对于 “报警组”、 “组变量”及常规标记名而言是有效的。
.NumAlarms	显示 “分布式报警对象”中的报警总数。
.PendingUpdates	指出 “报警显示对象”是否有任何待处理的更新。
.ROCPct	可读 / 写点域，用于监视和 （或）控制报警检查的变化率。
.ROCStatus	只读离散型点域，用于确定是否存在指定标记名的 “变化率”报警。
.SuppressRetain	读取 / 写入分布式报警显示对象的抑制保持状态。

下列报警点域与报警对象 （“分布式报警显示”）有关，而与标记名无关。

.AlarmClass	**.AlarmAccess**	**.AlarmDate**	**.AlarmGroupSel**
.AlarmLimit	**.AlarmName**	**.AlarmOprName**	**.AlarmOprNode**
.AlarmPri	**.AlarmProv**	**.AlarmState**	**.AlarmTime**
.AlarmType	**.AlarmValue**	**.Freeze**	**.ListChanged**
.NumAlarms	**.PendingUpdates**	**.SuppressRetain**	

这些报警点域通过下面的函数进行访问：

```
GetPropertyM(ControlName.Property, MsgTag)
```

运行期间执行时，从 **MsgTag** 获取属性，以便在选定的行中显示标记名。如果选定多个行，则 **MsgTag** 的属性是其中第一行中的标记名。

如需有关报警点域的详细信息，请参阅联机 *InTouch 参考指南*。

报警可见性控制

出于各种原因，有时您可能需要“关闭”某些报警，但并不实际删除某个项目的报警配置。InTouch 支持三种基本类型的报警可见性控制：**禁用**、**禁止**及**抑制**。这些会在下面的三阶段报警模型中作介绍。

- **报警刺激**：此时会跟踪*应报警*项目的值或状态。
- **报警状态**：应报警项目与报警限及条件作比较，以确定它是否处于*报警*状态。此时会跟踪与报告状态转换。
- **报警通知**：报警状态的报告与更新会通过一个或多个报警客户端进行*显示*和（或）*记录*。

在这个模型的基础上，“报警供应器”可处理下列可见性控制：

- **报警禁用**：“报警供应器”可**禁用**报警，方法是给它设置**禁用**的标记（例如状态位）。此种方法不涉及报警配置的其它更改。禁用某个报警时，可视具体情形确定是否继续检查会让该项目进入*报警*状态的条件。但是在禁用生效时，这些条件无法让该项目进入*报警*状态。实际上，该项目处于“强制正常”状态。

  禁用会切断**报警刺激**与**报警状态**之间的联系。

  - 请注意，禁用完全发生在“报警供应器”内部。
  - 所有的“报警接收器”均会看到同样的结果。
  - 因为在禁用某个报警时，该报警无法进入*报警*状态，所以报警历史中不会为该报警作任何记录。

- **报警禁止**：报警可以通过用某个标记名将它标示为禁止而加以**禁止**。此标记名被称为*约束标记*。此种方法不涉及报警配置的其它更改。在约束标记名为 FALSE（零或 NULL）时，会按正常方式处理该报警，但在约束标记名为 TRUE（非零或非 NULL）时，该项目将无法生成报警。就其本质而言，其作用与**禁用**报警相同。此时，我们便称该报警被**有效禁止**。很明显，报警禁止包括两个阶段：

  1. 指定约束标记名。
  2. 将约束标记名的状态从 FALSE 更改为 TRUE，或从 TRUE 更改为 FALSE。

与禁用相同，**禁止**也会切断**报警刺激**与**报警状态**之间的联系。

备注 将标记名指定为某个报警的约束标记名会增加其交叉引用使用计数。

在三阶段报警模型的基础上，“报警接收器”可处理下列可见性控制：

- **报警抑制**：“报警接收器”可通过设立一套排除标准来**抑制**一个或多个报警。如果某个报警符合排除标准，则在“报警接收器”中它是不可见的。这就是说，它不会出现在在该特定“报警接收器”的显示中，也不能打印或记录，具体取决于其功能。报警的实际生成完成不受抑制的影响。实际上，抑制会导致“报警接收器”忽略特定的一些报警。

  抑制报警会中断**报警状态**和**报警通知**之间的联系。

  - 请注意，抑制完全在“报警接收器”内部进行。

- 每个 “报警接收器”会看到不同的结果。这就是说，“报警接收器”可以有自己的一套排除标准，因此可以将不同的报警组设置为隐藏。实际上，抑制是对 “报警查询”的细化。
- 因为报警的实际生成并不会受到影响，所以即使一个或多个 “报警接收器”抑制报警，该报警的项目仍可记入报警历史（假定 Alarm DB Logger 自身也不抑制该报警）。

每次 “报警接收器”更改其抑制标准时， Logger 均会进行记录。

备注 如果在 alarmbuf.ini 中添加一行 "SuppressionLog=1"，则 "SuppressionLog=0" 会关闭抑制标准改变时写入 Logger 的功能。缺省条件下，此值设置为关闭。

运行时的禁用与禁止状态转换

InTouch 可供一次禁用或启用某个标记名的所有报警。此外，对于包含子状态的报警，每个子状态也可单独禁用。因此，所有的子状态均可禁用，或者是禁用一些子状态，而其它子状态仍保持为启用状态。例如，模拟 “值”报警可以启用 "Hi" 而禁用 "HiHi"。

与禁用的情况相同，对于包含多个子状态的报警，您也可以给各个子状态分别指定约束标记名。每个子状态均可使用不同的标记名进行禁止，您可以留一些子状态不给其指定约束标记名。例如，模拟 “值”报警的 "Hi" 值可使用一个标记名禁止、 "HiHi" 值可使用另一个标记名禁止，而 "Lo" 与 "LoLo" 则不予以禁止。

InTouch 只允许在 WindowMaker 中更改约束标记名。

在 WindowViewer 中，每当转换导致报警脱离**有效禁止**状态，便会执行检查逻辑来确定 “报警供应器”是否应将项目放入***报警***状态。

运行期间，“报警供应器”不会为**禁用**或**有效禁止**的报警或子状态生成报警。您可以在运行时更改报警的禁用或启用状态。更改约束标记名的值也可以在运行时进行。不过，给报警指定约束标记名的工作只能在 WindowMaker 中进行设置或更改。

报警或子状态可以单独禁用、禁止，或同时禁用与禁止。如果报警当前被禁用或有效禁止，则该报警会被 “关闭”。只有在既未被有效禁止又同时被启用的情况下，该报警才能够变为活动状态。因此，即使启用了报警，只要它被另一标记有效禁止，则该报警仍不能从供应器进入接收器。

如果报警或子状态没有被指定约束标记名，则效果同约束标记名始终为 FALSE 一样，也就是说，该项目永远不会被有效禁止。

只要报警从**禁用**转到**启用**，便会执行检查逻辑来确定 “报警供应器”是否应将该项目放入***报警***状态。

如果报警变为**禁用**或**有效禁止**，而项目仍处于***报警***状态，则该项目会被强制转换为一个不同 （有效）的状态。具体该状态会是什么，这取决于有哪些可用的状态，以及它们是否也已经被禁用。此活动由 “报警供应器”根据报警类型、极限值等进行处理。

被禁用或有效禁止的报警不会等待*确认*。如果报警包含子状态，则它只能等待*确认*仍可用的子状态。

例如：

作为一个简单的**禁用**示例， InTouch 应用程序工程师可以通过设置控件或脚本给标记名设置适当的点域：.AlarmDisabled 用于禁用报警，.AlarmEnabled 用于启用它。此点域也可以在外部应用程序中控制，方法是通过 NetDDE 或 SuiteLink 使用 InTouch 作为 I/O 服务器。

如需有关报警点域的详细信息，请参阅联机 *InTouch 参考指南*。

作为一个**禁止**示例， InTouch 应用程序工程师可以设置一个标记名 "Phase3"，在特定的执行阶段该标记名会变为 True。对于应用程序中的特定报警，工程师可以将 Phase3 指定为约束标记名。这表示，在 Phase3 为 True 时，这些报警会被禁用，在 Phase3 为 False 时，这些报警则可以正常工作。

配置报警系统

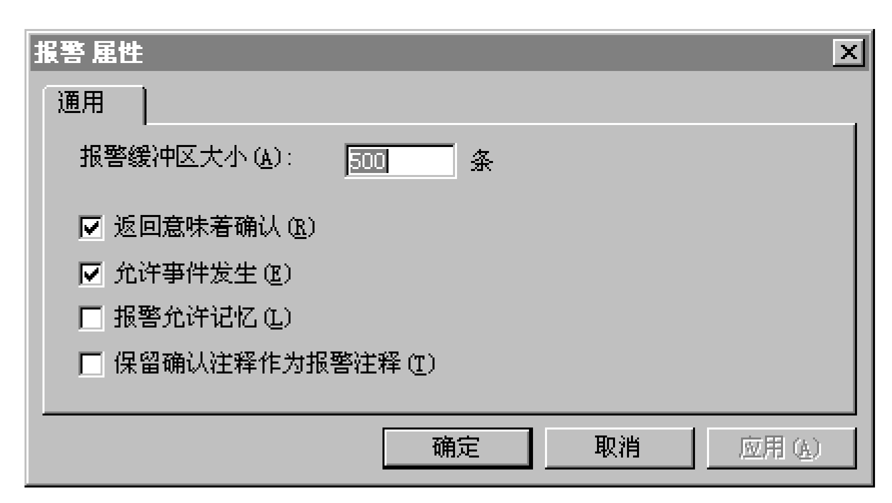
您可以配置报警系统的各种参数，如是否启用事件、在报警恢复正常时是否需要确认等等。

备注 配置对话框与所有的标准 Windows 属性页均很相似，在单击**确定**之前不会记录任何设置。如果单击**取消**，则会忽略所有输入并关闭对话框。

报警 / 事件通用属性

要配置报警 / 事件通用属性

1. 在**特别**菜单上，指向**配置**，然后选择**报警**，或者在 “应用程序浏览器”中的**配置**下，双击**报警**。此时会出现**报警属性**对话框：



提示 如果右击任何一个报警配置对话框中的文本框，则会出现一个弹出式菜单，显示可应用于所选文本的各条命令。

2. 在**报警缓冲区大小**框中，输入要让 WindowViewer 保存的 “内存” 报警事件数。（节点可以为摘要或历史查询存储的最大报警数目）。

备注 只有 “内存” 报警事件才会显示在报警显示对象中。如果未使用报警，则此值可设置为 1 以节约内存。

如果将此值设置得太高，则会降低系统性能。对于 “分布式报警系统”，建议使用 500。

3. 如果希望自动确认已恢复 “正常” 状态 (RTN) 的报警标记名，请选择**返回意味着确认**。如果在报警返回正常状态后仍希望操作员去确认它，请不要选择此选项。

备注 使用**扩展的摘要**确认模型定义标记名时，**返回意味着确认**选项不会应用于该标记名。

4. 对于操作员、 I/O、 QuickScript 或系统引起的所有数据变化，如果希望打开其事件记录功能，请选择**事件已启用**。 （只有已选取**记录事件**的标记名会受到影响）。

   如需有关事件的详细信息，请参阅 “报警与事件”。

5. 如果希望在 WindowViewer 关闭时保留 **.AlarmEnabled** 变量的状态，请选择**启用报警记忆**。

6. 如果希望保留确认报警时输入的注释并以之更新标记名的 “报警注释” 点域，请选择**保留确认注释作为报警注释**。在操作员确认报警时，有一个选项可用来提供有关该报警的注释。确认注释也可以通过脚本函数来提供。如果选中 “保留确认注释作为报警注释” 框，则确认注释会被保留并用作相应标记名的报警注释。InTouch 还会更新标记名字典中的 “报警注释”。

   如需有关详细信息，请参阅 “给报警确认函数附加注释”。

7. 单击**确定**以保存设置并关闭对话框。

给报警确认函数附加注释

现在添加了一个新的点域 AlarmComment，专门用于给报警提供注释。AlarmComment 最长可达 131 个字符。

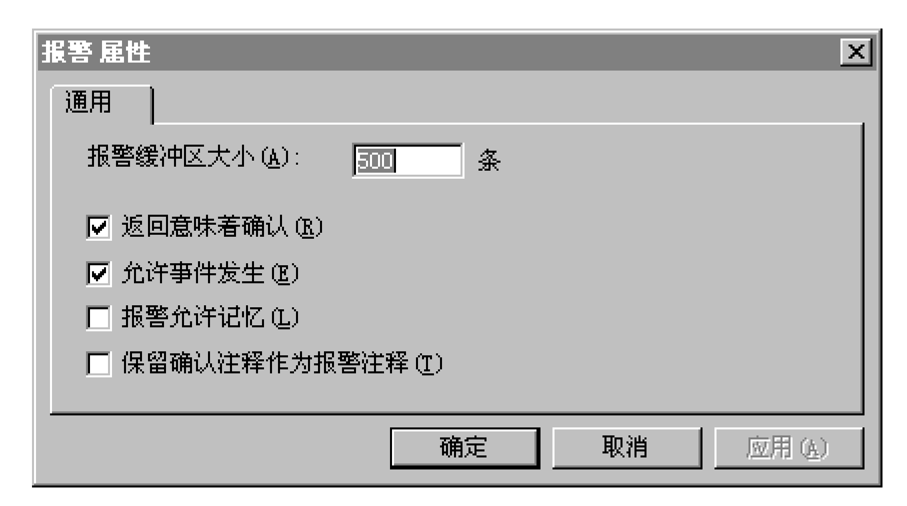
每个报警确认函数均可以附加一条注释 - 无论确认是通过 “分布式报警对象”、脚本函数还是以任何其它方式进行的。确认报警的操作员可以使用此注释来添加有关报警的信息。

在报警变为活动状态时，“分布式报警系统”会创建一条报警记录来跟踪该报警实例。对于与报警起因有关的注释， InTouch 使用数据库中标记名报警定义中的 “报警注释”框。如果操作员在确认报警时提供注释，则 InTouch 会将它用作该报警实例的 “确认注释”与 “分布式报警系统”的注释添加到 “报警记录”中。这两者均会记录到报警数据库中。 “分布式报警对象”和 “报警打印程序”会根据报警实例是否已获得确认来分别显示 “报警注释”或 “确认注释”。下一次相同的标记上发生报警时，该 “报警注释”会再次被用作新报警实例的起因。在确认新实例时，操作员可以输入一则不同的 “确认注释”。

您也可以将 InTouch 配置为 “保留确认注释作为报警注释”，以使用“确认注释”来更新标记名数据库中的 “报警注释”（此功能曾用于更新标记名注释，但现在它用于更新 AlarmComment）。如果启用此项功能，则运行期间 AlarmComment 点域会被改写 - 包括 “标记名数据字典”中的 AlarmComment 项。

要使用报警确认注释来更新 AlarmComment 字段

1. 在**特别**菜单上，指向**配置**，然后选择**报警**。此时会出现**报警属性**对话框，并显示**常规**属性页。



提示 如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 如果希望保留确认报警时输入的注释，以之更新相应标记名的 AlarmComment 点域，并将它复制到 “标记名字典”，请选择**保留确认注释作为报警注释**。如果未选中此复选框，则 “确认注释”仍会同已确认的报警一起显示 （在数据库、打印件及显示画面中），但 “报警注释”将保持不变。

   如需有关报警注释的详细信息，请参阅第 6 章 “标记名字典”。

3. 单击**确定**。

显示报警统计

“分布式报警系统”提供一个内置的报警统计对话框。应用程序开发人员可以将应用程序设计成调用**报警统计**对话框，以列出特定报警显示当前查询的状态。此对话框还可以通过从快捷菜单选择**统计**来激活。

如需有关详细信息，请参阅 “选择与配置报警查询收藏夹”。

报警统计对话框为您提供特定报警显示的当前报警查询的概况。它列出实际报警供应器的请求与每个请求返回的结果。特别需要注意的是，即使您可能只要求提供一个 “报警组列表名”，该名称仍可能对应多个 “报警供应器”查询。例如：



提示 对话框中的每行均列出数字与查询。该数字表示已返回的查询百分比。对话框提供查询结果的静态显示。

要更新查询列表中检索到的报警百分比

1. 右击分布式报警对象。
2. 从子菜单中选择**统计**。
3. 选择要更新的报警查询，然后单击**更新**。
4. 单击**确定**以关闭对话框。

确认本地报警

要确认本地报警，您可以在动作或键 QuickScript 中使用 **.Ack** （点域），或者在 “分布式报警对象”上使用上下文相关右击菜单。

备注 对于扩展的报警摘要，不建议使用。

要创建本地报警确认按钮

1. 创建一个三维按钮或任何其它对象，将它链接到动作或键 QuickScript。
2. 双击对象，或选择它，然后在**特别**菜单上选择**动画链接**。

3. 在“动画链接”选择对话框的**触动按钮**区，单击**动作**。此时会出现 QuickScript 编辑器。

   给 QuickScript 输入下列任何语句：

语句	描述
Ack $System;	确认系统中的所有本地报警。
Ack Group Name;	确认特定“报警组”中的所有本地报警。
Ack Tagname;	确认特定标记名的报警。
$System.Ack=1;	确认系统中的所有本地报警。
Group Name.Ack=1;	确认特定“报警组”中的所有本地报警。
Tagname.Ack=1;	确认特定标记名的报警。
Tagname.AckDev=1;	确认特定标记名的偏差报警。
Tagname.AckROC=1;	确认特定标记名的变化率报警。
Tagname.AckValue=1;	确认特定标记名的值报警。

4. 单击**确定**。

如需有关 QuickScript 编辑器及其功能的详细信息，请参阅第 8 章“在 InTouch 中创建 QuickScript”

分布式报警系统中的时间标签

目前“分布式报警系统”的“报警记录”有 5 个时间标签：

时间标签	描述
ar_OrigTime	发生时间，即发生报警的时间，简写为 OAT。
ar_SubTime	最近子状态发生更改的时间，如从 VALUE/HI 到 VALUE/HIHI
ar_AckTime	最近确认报警的时间
ar_Time	上次发生更改的时间，简写为 LCT
ar_RtnTime	返回正常状态的时间，简写为 RTN。

从旧的 InTouch 标准报警系统迁移到分布式报警系统

InTouch 可以自动迁移使用“标准报警对象”的 InTouch 应用程序。您试图在 WindowMaker 中打开包含“标准报警对象”的任何应用程序时，InTouch 会使用缺省值自动将所有的“标准报警对象”转换为“分布式报警对象”。但颜色、字体、表达式以及报警查询等设置不会迁移。

您试图在 Window Viewer 中打开使用 7.11 之前的版本创建的应用程序时，会弹出一个对话框，提示您运行 WindowMaker 来转换该应用程序。如果该应用程序中包含 “标准报警对象”，则转换过程会将 “标准报警对象” 转换为 “分布式报警对象”。

如果单击**是**，则应用程序会在 WindowMaker 中打开并进行转换。如果单击**否**，则 Viewer 会关闭。

从旧的主 / 从报警系统迁移到分布式报警系统

如 “从旧的 InTouch 标准报警系统迁移到分布式报警系统” 部分曾提到的，“主 / 从” 应用程序中所有的 “标准报警显示” 均会为 “分布式报警显示” 所取代。新的 “分布式报警显示” 将使用确认查询 "\\nodename\intouch!$system"，其中的节点名是主节点的名称。

您现在应能够使用分布式报警显示来查看和确认远程报警。

通过对 .Ack 点域赋值来确认远程报警的操作与以往相同。并不需要更改应用程序。根据 I/O 标记名是为 NetDDE 还是 SuiteLink 配置的，有可能需要启用 NetDDE。不过，由于现在可通过 “分布式报警显示” 来确认报警，您可能会决定不再需要用单独的控件发出确认。

通过提示 .Alarm 点域来监视报警状态的操作也与以往相同。并不需要更改应用程序。根据 I/O 标记名是为 NetDDE 还是 SuiteLink 配置的，有可能需要启用 NetDDE。

热备份与同步

“InTouch 分布式报警系统” 为 “报警供应器” 应用程序 （如 InTouch）提供了通过网络发出报警通知及接收报警确认的功能。它也为远程节点上的 “报警接收器” 应用程序 （客户端）提供了查询、显示与确认这些报警的功能。在分布式报警环境下，可能有许多 “报警供应器” 与 “报警接收器” 通过网络连接在一起。

此外，有可能将某些 “报警供应器” 配置为系统中其它 “报警供应器” 的***备份***。使用备份 “报警供应器” 的目的是建立一个安全保障系统 - 在*主* “报警供应器” 出现故障时，***辅助*** “报警供应器” 能够接管其报警通知工作，从而确保在某些关键环境下能够正常生成报警。例如，您可以设置两个独立的计算机节点，让它们运行同样的 InTouch 应用程序，与相同的设备连接，甚至还可能使用一组冗余的 I/O 连接。或者，主 “报警供应器” 可能是一个 “全功能” 的应用程序，而辅助 “供应器” 则是一个 “微型” 系统，只处理最关键的部分，以确保设备正常工作和免受损坏。通常，只有一些关键的 “报警供应器” 才会采用辅助 “报警供应器” 作为备份。

要设置备份配置，您可以定义一个***热备份对***，给它指定一个***名称***，并分别确定*主* “报警供应器” 与***辅助*** “报警供应器”。

备注 只有在主供应器处于不活动状态时，才会显示来自辅助供应器的报警。

备份配置的基础是 “报警接收器”（客户端）使用一个引用*热备份对*的名称的报警查询。对于该*热备份对*，客户端只看到一组报警：即来自主供应器的报警，如果主供应器出现故障，则是来自辅助供应器的报警。此外，主供应器与辅助供应器上的报警确认完全*同步*。这就是说，如果报警在主 “供应器”上得到确认，则在辅助 “供应器”它同时也会被确认。

“InTouch 分布式报警系统”的 “热备份”与同步功能包括：

1. 提供一个配置实用程序，可简化建立报警备份对、指定主供应器与辅助供应器的过程。
2. 提供一个配置实用程序，可执行*报警映射*。
3. 内置报警确认同步功能。
4. “报警供应器”启动 / 关闭期间的同步交换功能。

有关热备份对的注意事项

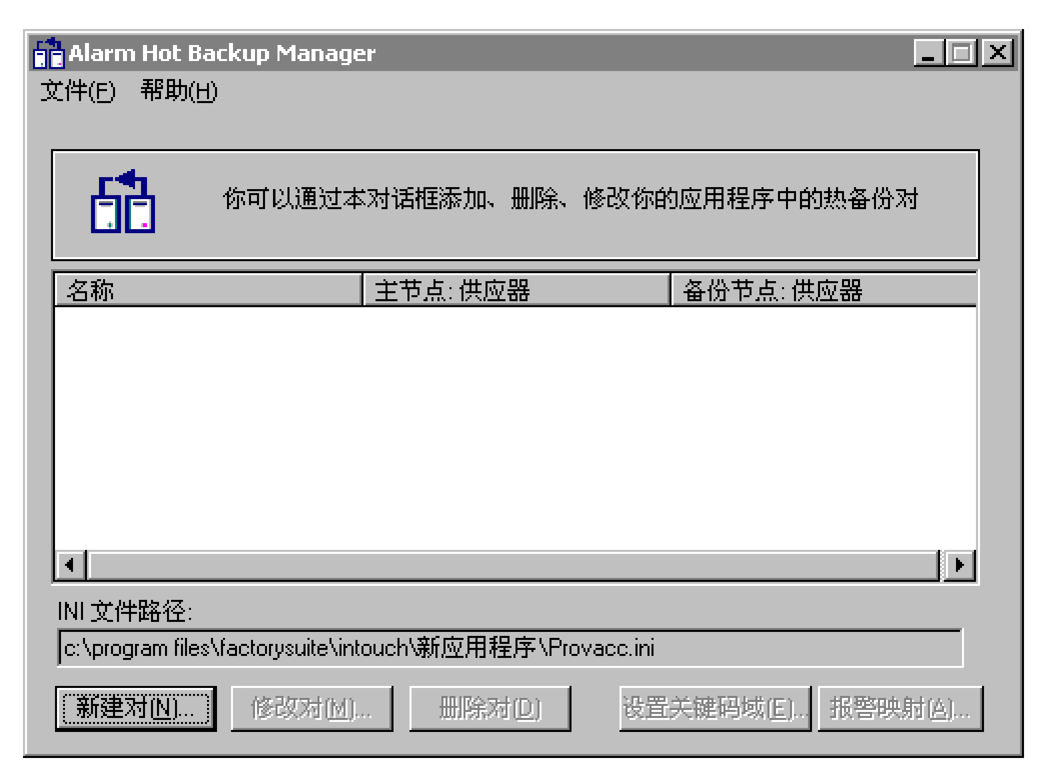
1. “热备份”仅支持 InTouch 7.11 及更高版本，不支持其它客户端。
2. “扩展的报警摘要”或面向事件的报警不支持 “热备份”。
3. 如果您让报警客户端 （“分布式报警”显示对象）查询 “热备份”，随后又让它再次单独查询该备份对的 “主供应器”，则 “分布式报警”显示对象会显示重复的记录。
4. “供应器”不得配置为一个以上 “热备份”对为主供应器或辅助供应器。
5. 如果在确认主 “供应器”的记录之后，再次出现辅助 “供应器”（确认发生时刚好宕机）的记录，则辅助 “供应器”已确认记录的时间标签将是辅助 “供应器”上的时间标签，而不再与主 “供应器”的完全相同。
6. 查询 “热备份”对的报警客户端 （“分布式报警”显示对象）会在相应的列中显示对名作为 “供应器”，而不会显示 “供应器”的节点名。
7. 您可以选择**设计时**与**运行时**报警记录字段的任意组合来进行映射。不过，切记要确保这些映射不会导致多重引用。
8. 映射**值**与**极限**等关键字段时，数值会被舍入为四位小数，然后再进行映射。
9. 没有特定的设计时与运行时映射组合的报警记录将使用缺省的运行时映射。
10. 充当 “热备份”客户端的本地节点不得配置为 “热备份供应器”之一。
11. “报警供应器热备份”对不得将已配置为客户端的节点用作 “报警供应器”之一。“热备份”实用程序旨在处理这样的热备份对：其中的 “主报警供应器”和 “辅助报警供应器”与在其中进行配置的系统均不相同。

热备份配置

“热备份同步”实用程序负责所有必要的 “热备份”配置。在安装 InTouch 时，“热备份”实用程序会自动添加到 WindowMaker 的 “应用程序浏览器”中。

要配置热备份

1. 启动 WindowMaker。
2. 在 “应用程序浏览器”中双击 Alarm Hot Backup Manager （报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager** （报警热备份管理器）对话框。



备注 缺省条件下，此实用程序会自动检查上次打开的 InTouch 应用程序文件夹中的 provacc.ini 文件。如果存在该文件，则实用程序会使用该文件进行配置。否则，该实用程序会在所需的文件夹中创建一个新文件。

要更改缺省的 INI 文件

1. 启动 WindowMaker。
2. 在 “应用程序浏览器”中，双击 Alarm Hot Backup Manager （报警热备份管理器）图标。此时会出现 **Alarm Hot Backup Manager** （报警热备份管理器）对话框。

3. 在**文件**菜单上，选择**打开**命令。此时会出现**打开**对话框，列出所有可用的 provacc.ini 文件。（此对话框只允许您选择一个 provacc.ini 文件）。

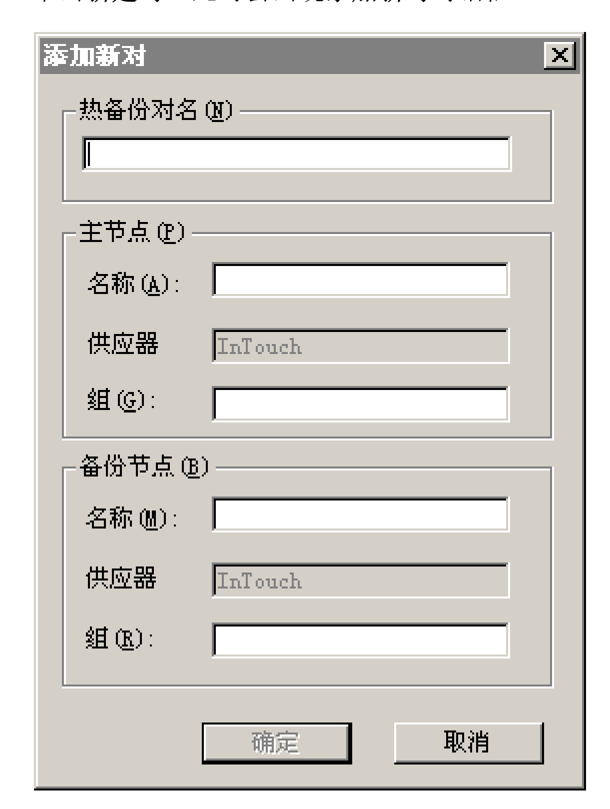


4. 如有必要，请更改路径，然后单击**打开**。（建议将 provacc.ini 文件放在应用程序目录中）。

要添加新的热备份对

1. 启动 WindowMaker。
2. 在“应用程序浏览器”中，双击 **Alarm Hot Backup Manager**（报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager**（报警热备份管理器）对话框。

3. 单击**新建对**。此时会出现**添加新对**对话框。



4. 在**热备份对名**框中，给新备份对输入唯一的名称。备份对名最长可使用 32 个字母数字字符。

备注 备份对名不允许重复。下划线 (_) 是唯一允许用在 "热备份" 对名开头、中间或结尾的特殊字符。

5. 在**主节点**组的**名称**框中，输入运行主供应器应用程序的计算机的名称。节点名最长可使用 32 个字母数字字符。

   对于节点名：

   - 节点名必须以字母打头
   - 可使用任意字母数字字符
   - 唯一允许使用的特殊字符是下划线 (_)
   - 节点名中的任意位置均不允许使用其它特殊字符

备注 特定应用程序的节点名在整个 "热备份" 列表中必须保持唯一。如果输入了错误或重复的项目，则会出现相应的错误消息框。

备注 InTouch 是唯一的供应器名，并因此不得更改。

6. 在**主节点**组的**组**框中，输入从主供应器查询报警的 "报警组" 的名称。例如 $System。"报警组" 的名称最长可使用 32 个字母数字字符。

   对于组名：

- 允许使用所有有效的字符（ A 到 Z、a 到 z、0 到 9、!、@、-、?、 #、 $、 %、 _、 \、 &）。
- 组名必须以 A 到 Z 或 a 到 z 打头。
- $ 是唯一可作为组名开头字符的其它字符，如果确有使用，则该组名只能是 $System。
- 不得在组名结尾使用反斜杠 (\)。

重要！ 现使用 "\" 字符而不是 "!" 字符来区分对名与组名。

- 不允许使用双反斜杠 (\\)。

7. 在**备份节点**组的**名称**框中，输入运行备份供应器应用程序的计算机的名称。节点名最长可使用 32 个字母数字字符。
8. 在**备份节点**组的**组**框中，输入从备份供应器查询报警的 “报警组” 的名称。例如 $System。“报警组”的名称最长可使用 32 个字母数字字符。

备注 热备份对名不得是另一个热备份配置中的节点。

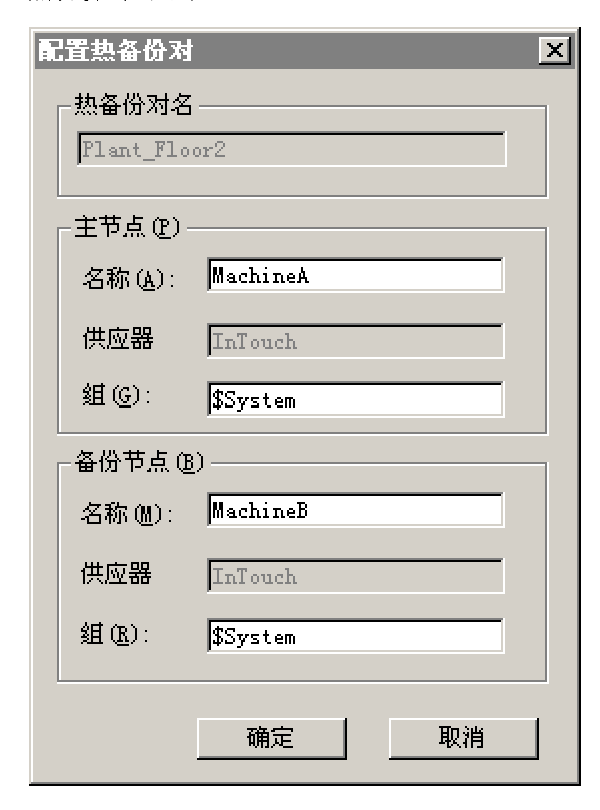
9. 单击**确定**。

备注 要将所有更改保存到 INI 文件，必须执行**文件**菜单上的**保存**命令。在退出该实用程序之前，它会提示您保存更改。

要编辑现有的热备份对

1. 启动 WindowMaker。
2. 在 “应用程序浏览器” 中，双击 **Alarm Hot Backup Manager** （报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager** （报警热备份管理器）对话框。

3. 选择要修改的 “热备份对名” , 然后单击**修改对**。此时会出现**配置热备份对**对话框。



备注 特定应用程序的节点名在整个 “热备份” 列表中必须保持唯一。如果输入了错误或重复的项目，则会出现相应的错误消息框。

对于节点名：

- 节点名必须以字母打头
- 可使用任意字母数字字符
- 唯一允许使用的特殊字符是下划线 (_)
- 节点名中的任意位置均不允许使用其它特殊字符

对于供应器名：

- InTouch 是唯一的供应器名，并因此不得更改。

对于组名：

- 允许使用所有有效的字符（ A 到 Z、 a 到 z、 0 到 9、 !、 @、 -、 ?、 #、 $、 %、 _ 、 \、 &）。
- 组名必须以 A 到 Z 或 a µ ? z 打头。
- $ 是唯一可作为组名开头字符的其它字符，如果确有使用，则该组名只能是 $System。
- 不得在组名结尾使用反斜杠 (\)。
- 现使用 "\" 字符而不是 "!" 字符来区分对名与组名。

- 不允许使用双反斜杠 (\\)。

如需有关此对话框中各个字段的详细信息，请参阅 "要添加新的热备份对" 过程。

4. 编辑必要的字段，然后单击**确定**。

备注 要将所有更改保存到 INI 文件，必须执行**文件**菜单上的**保存**命令。在退出该实用程序之前，它会提示您保存更改。

要删除现有的热备份对

1. 启动 WindowMaker。
2. 在 "应用程序浏览器" 中，双击 **Alarm Hot Backup Manager** （报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager** （报警热备份管理器）对话框。
3. 选择要删除的 "热备份" 对，然后单击**删除对**。此时会出现一个消息框，要求您确认删除。
4. 单击**是**。

备注 要将所有更改保存到 INI 文件，必须执行**文件**菜单上的**保存**命令。在退出该实用程序之前，它会提示您保存更改。

设置报警记录的关键码域

要实现同步，您必须确定报警记录域组合，使这些域的组合值可生成一个唯一的映射关键码。组合映射关键码域可以是设计时与运行时的组合。

要选择报警记录关键码域

1. 启动 Alarm Hot Backup Manager （报警热备份管理器）实用程序。

提示 您可以通过在 WindowMaker 的 "应用程序浏览器" 中双击它来启动。

2. 选择 “热备份对”，然后单击**设置关键码域**。此时会出现**选定关键码域**对话框。

3. 在**报警记录域**组中，选择要包含到映射关键码列表的域。（选定的域会添加到**已选定的域**列表框中）。

4. 如果某个报警记录域的可能值在应用程序设计时不同并且已知，请为这个域选择**设计时**选项。例如，在 InTouch 供应器的情况下，“名称”域的值在设计时已知，因为它会采用主节点与备份节点应用程序中定义的标记名的值。缺省条件下，**报警名**与**运行时**是预先选定的。

备注 **报警记录域**可以在运行时或设计时配置。所有的设计时域均会转入任意标记名映射的**映射报警记录**对话框。

如需有关详细信息，请参阅 “映射报警记录”。

5. 如果某个报警记录域的可能值在主备份供应器与备份供应器中均相同，请为这个域选择**运行时**选项。缺省条件下，**报警名**与**运行时**是预先选定的。

备注 在运行时，如果*设计时域*是组合关键码的一部分，并且主供应器不能在映射表中找到该关键码的映射项，则 “热备份”会在备份供应器中自动查找使用相同标记名的报警，并发送相应报警的确认，反之亦然。

---

6. 单击**确定**。

备注 要将所有更改保存到 INI 文件，必须执行**文件**菜单上的**保存**命令。在退出该实用程序之前，它会提示您保存更改。

映射报警记录

只要主报警供应器与辅助报警供应器并非完全相同的应用程序，您都需要映射 “报警供应器”。例如，在主供应器是 “全功能”的应用程序，而辅助供应器是 “微型”应用程序时，报警的名称可能会不同。 “报警供应器”映射会在主供应器与辅助供应器之间建立一种通讯机制，以便在某个供应器上确认报警时，“分布式报警系统”知道要在其它供应器上确认的报警。

备注 映射是从一个 .csv （逗号分隔变量）文件导入的，该文件需要使用简单的文本文件编辑器或 Microsoft Excel 来创建。报警域名必须与**映射报警记录**对话框包含的名称匹配。否则，导入过程会中止并显示一个警告消息框。

每个 “热备份”对可以由应用程序级的唯一名称确定。例如：

热备份对名	主节点 ! 供应器	备份节点 ! 供应器
BoilerRoom1	NodeA!InTouch	NodeX!InTouch

备注 为了简化您的工作，如果未提供映射，“分布式报警系统”会假定主供应器与辅助供应器上相同的报警之间存在映射关系。

您可以合并报警记录的域值 （如报警组、报警名及优先级），以生成一个唯一确定报警记录的 “组合映射关键码”。

InTouch “报警供应器”会给 “名称”域赋予生成报警的标记名的名称。因此，在给定 “热备份对”的情况下，会使用组与 “名称”域的组合来生成一个映射关键码。

例如：

供应器节点	备份节点
$System!TagA	$System!TagB

如果某个供应器将名称域与注释域用作一个唯一的域，则映射关键码可以是名称与注释的组合。

供应器节点	备份节点
tagA!CommentA	tagB!CommentB

对于第三方供应器的任何其它域组合，此方法也适用。

在 Alarm Hot Backup Manager （报警热备份管理器）实用程序的 “选定关键码域”对话框与 .csv 文件 （用于在主报警供应器与辅助报警供应器之间创建报警映射的 .csv 文件）的标题名中，使用的术语有所不同。下表概括了其间的关系：.

“选定关键码域”对话框	.csv 标题名
报警组	组
报警名	名称
报警类	类

报警类型	类型
优先级	优先级
报警值	值
报警限	报警限
注释	注释

备注 创建用于导入报警映射的 .csv 文件时，应该将右边显示的术语用作标题名。

重要！ .csv 文件中不能使用 "~" 进行报警映射。

重要！ 使用 Excel 创建 .csv 文件时，不要在单元格中包含 ','，这是因为在创建 .csv 文件时， Excel 会自动生成此字符。

导入文件所需的详细信息

下列情况会使得文件无法导入：

- 所需的列均必须填入导入文件中所有记录的值。任何记录都不得缺少或多出一些值。
- 导入文件的标题应该与 “映射报警记录”对话框的标题相同，并且应遵循相同的顺序。

在下列情况下，如果导入的记录有误，则程序会提示用户忽略该特定的记录号，或者中止导入过程。

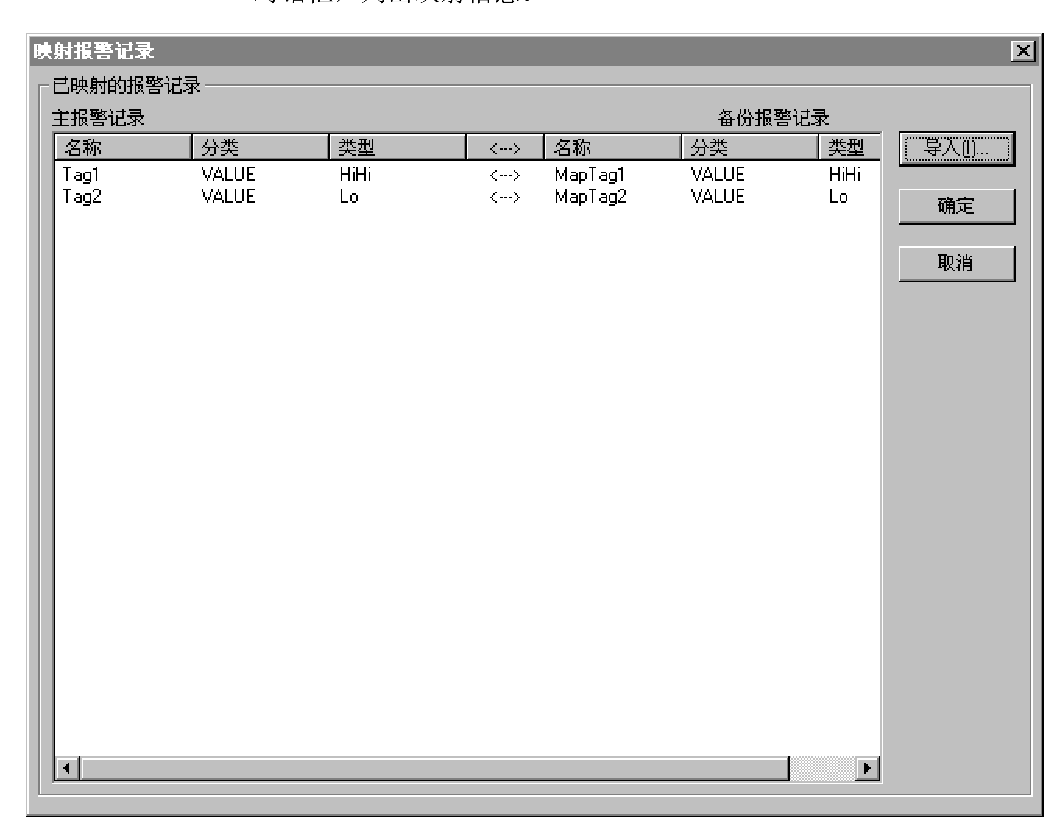
- “组”列的值不应包含空格。
- “名称”列的值不应包含空格。
- “类”列的值不得是 VALUE、 DEV、 ROC 以及 DSC 之外的任何值。
- “类型”列的值不得是 LOLO、LO、HI、HIHI、MinDev、MajDev、ROC 以及 DSC 之外的任何值。
- “优先级”列的值应该是从 1 到 999 的数字。
- 特定节点中的特定记录的 “类”或 “类型”的值未知时，“值”与“报警限”列的值可以是 Null 之外的任何内容。
- 在特定节点中的特定记录的 “类”值已知是 Value、 Dev 或 ROC 时，“值”与 “报警限”列的值只能接受 1234567890.-+eE 等字符。
- 在特定节点中的特定记录的 “类型”值已知是 LOLO、 LO、 HI、HIHI、 MinDev、 MajDev 或 ROC 时，“值”与 “报警限”列的值只能接受 1234567890.-+eE 等字符。
- 特定节点中的特定记录的 “类”或 “类型”的值已知为 DSC 时，“值”与 “报警限”列的值可以是 Null 之外的任何内容。
- “注释”列的值没有任何限制。
- 导入文件中不得存在重复的记录。如果存在重复的记录，则会忽略这些记录并在导入过程结束后向用户提供详细信息。

要映射报警记录

1. 启动 **Alarm Hot Backup Manager**（报警热备份管理器）实用程序。

   提示 您可以通过在 WindowMaker 的 “应用程序浏览器” 中双击它来启动。

2. 选择 “热备份对”，然后单击**映射报警**。此时会出现**映射报警记录**对话框，列出映射信息。



3. 单击**导入**。此时会出现**打开**对话框。
4. 找到并选择要导入的 .csv 文件。
5. 单击**确定**。

   备注 要将所有更改保存到 INI 文件，必须执行**文件**菜单上的**保存**命令。在退出该实用程序之前，它会提示您保存更改。在新导入期间，如果存在已经导入过并已显示在此对话框中的记录，则程序会提示用户是否要改写现有的这些记录。

   备注 在 “映射报警记录” 对话框中导入 CSV 之后，不会对 “报警类” 与 “报警类型” 等字段进行交叉验证。

热备份用法示例

本节介绍一个设置与实施 “报警热备份对” 的示例。

假设您在三台机器上安装了 InTouch。其中两台 （**MachineA** 和 **MachineB**）运行 “完全相同”的 InTouch “报警供应器”应用程序。**MachineA** 的应用程序可以生成两个名称分别是 **Tag1** 和 **Tag2** 的摘要报警。**MachineB** 的应用程序可以生成两个逻辑上完全相同的报警，分别为 **MapTag1** 和 **MapTag2**。

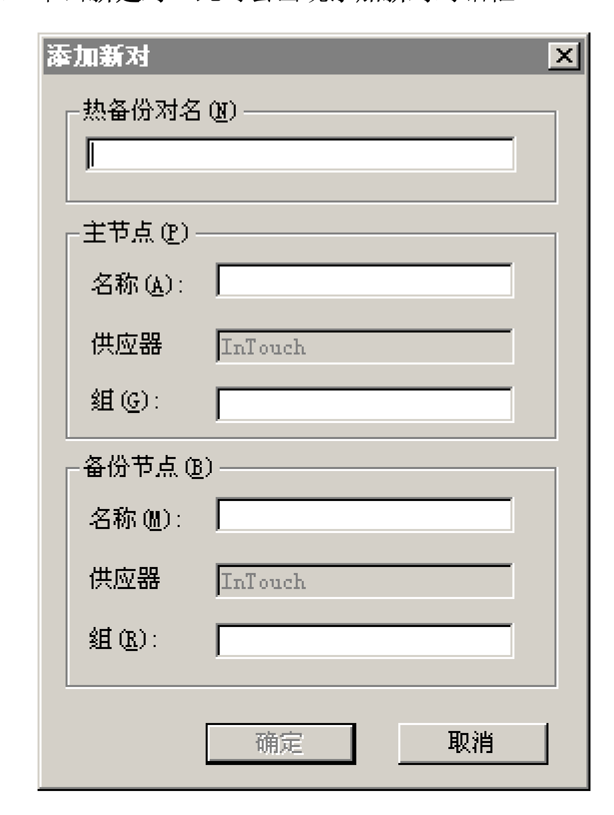
第三台机器则是一台为 “热备份”配置的 InTouch “报警接收器”。

要配置热备份对

1. 在客户端节点上创建一个新的 InTouch 应用程序。
2. 启动 WindowMaker。
3. 在 “应用程序浏览器”中，双击 **Alarm Hot Backup Manager** （报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager** （报警热备份管理器）对话框。



4. 单击**新建对**。此时会出现**添加新对**对话框。



5. 在**热备份对名**框中输入 **Pair1**。

6. 因为主“报警供应器”InTouch 应用程序运行在 **MachineA** 上并使用“报警组” **$System**，我们对**主节点**组的各字段作如下配置：

   - 主节点名称 = **MachineA**
   - 主节点供应器 = **InTouch**
   - 主节点系统 = **$System**。

7. 因为辅助“报警供应器”InTouch 应用程序运行在 **MachineB** 上并使用 “报警组” **$System**，我们对**备份节点**组的各字段作如下配置：

   - 备份节点名称 = **MachineB**
   - 备份节点供应器 = **InTouch**
   - 备份节点组 = **$System**。

8. 此配置完成之后，单击**确定**以保存新备份对。接下来我们要映射关键码域。

要映射报警记录关键码域

1. 在 “应用程序浏览器”中，双击 **Alarm Hot Backup Manager** （报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager** （报警热备份管理器）对话框。

2. 在列表中选择“热备份对”**Pair1**，然后单击**设置关键码域**。此时会出现**选定关键码域**对话框。



3. 因为我们通过报警名来映射报警，并且两个供应器的值不同，所以我们会选择**报警名**与**设计时**选项。

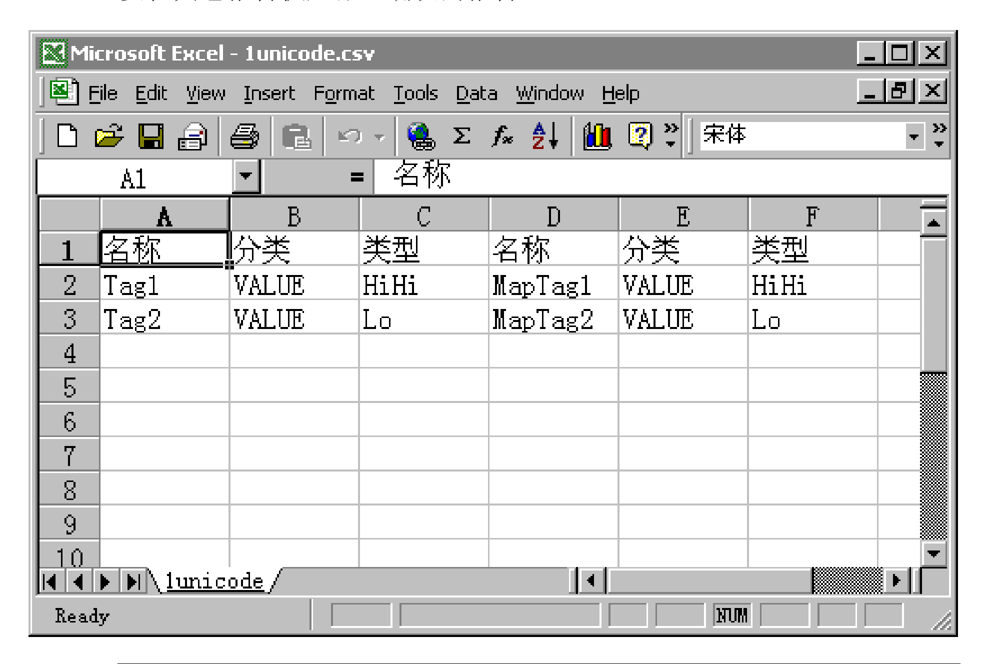
4. 接下来单击**确定**以保存配置。此时会出现**热备份配置**消息框，提示保存配置。

5. 单击**是**。现在我们就可以开始创建一个包含所有报警记录映射的 .csv 文件。

要创建报警映射 .CSV 文件

在本例中，我们在三台机器上安装了 InTouch。其中两台（**MachineA** 和 **MachineB**）运行 “完全相同”的 InTouch “报警供应器”应用程序。**MachineA** 的应用程序可以生成两个名称分别是 **Tag1** 和 **Tag2** 的摘要报警。**MachineB** 的应用程序可以生成两个逻辑上完全相同的报警，分别为 **MapTag1** 和 **MapTag2**。

因此，我们创建一个 .csv 文件 **pair1.csv**，让它包含使用 Microsoft Excel 创建的报警记录映射（映射是从这个 .csv 文件导入的）。报警记录映射可确保在一个“报警供应器”上确认报警时，“分布式报警系统”知道要在其它报警供应器上确认的报警。



备注 报警域名必须与**映射报警记录**对话框包含的名称匹配。否则，导入过程会中止并显示一个警告消息框。

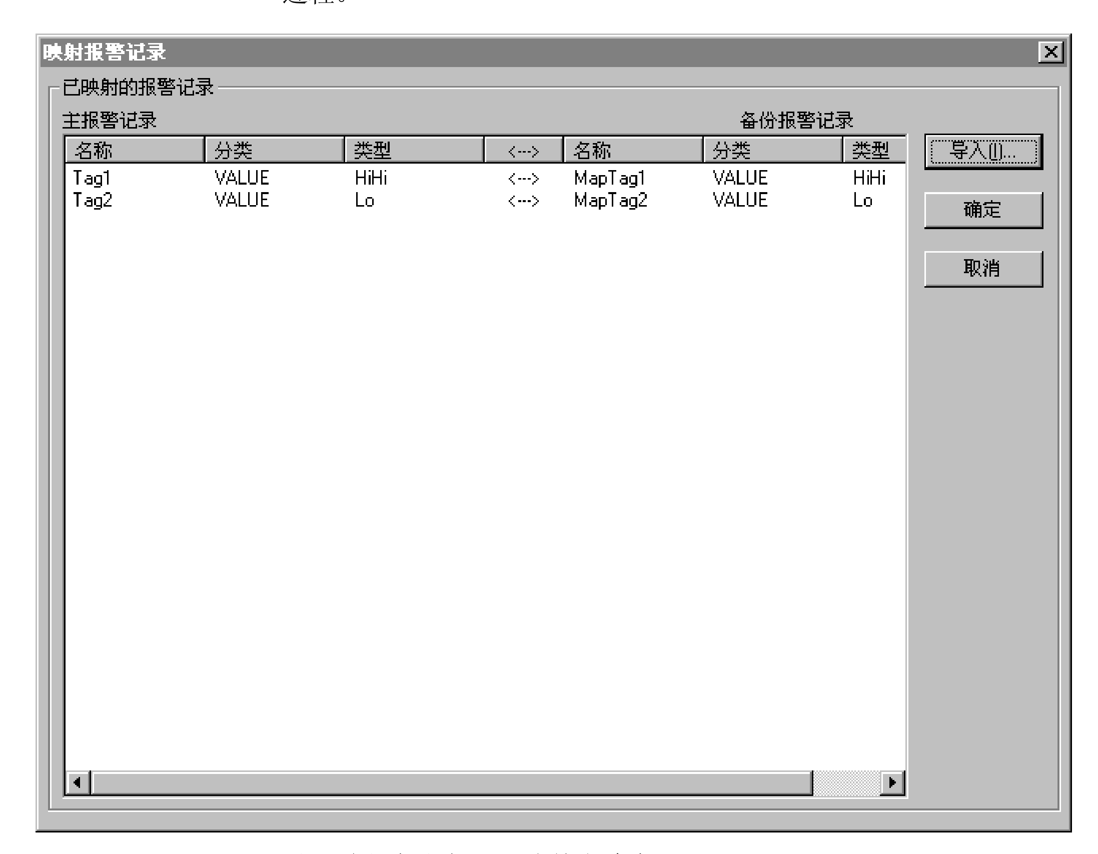
如需有关详细信息，请参阅“映射报警记录”。

现在我们已准备好导入这个包含所有报警记录映射的 .csv 文件。

要导入报警记录映射 .csv 文件

1. 启动 WindowMaker。
2. 在“应用程序浏览器”中，双击 **Alarm Hot Backup Manager**（报警热备份管理器）项。此时会出现 **Alarm Hot Backup Manager**（报警热备份管理器）对话框。
3. 在列表中选择 **Pair1**，然后单击**映射报警**。此时会出现一个空白的“映射报警记录”对话框。

4. 接下来单击**导入**。此时会出现**打开**对话框。找到并选择 .csv 文件 **(pair1.csv)**，然后单击**打开**。此时 “分布式报警系统”便开始导入过程。



5. 导入过程完成之后，请单击**确定**。

现在就可以运行 “热备份”应用程序了。

要运行热备份

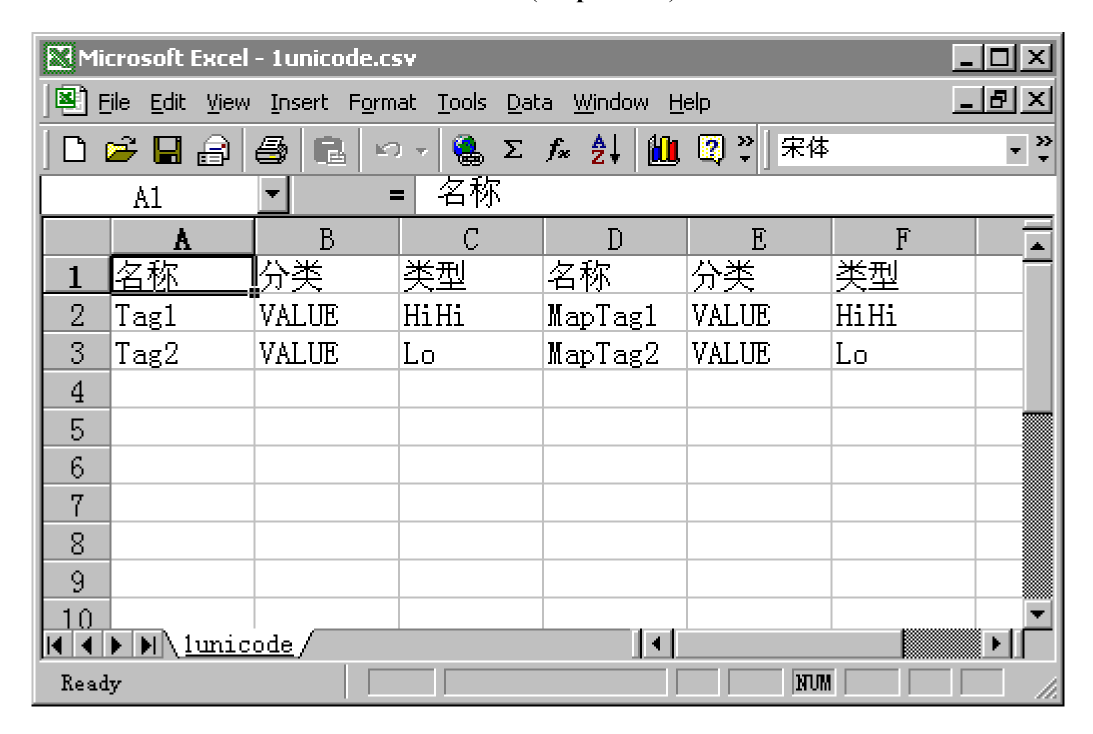
1. 启动两个 “报警供应器”应用程序。
2. 在运行时，生成报警 **Tag1**、 **Tag2**、 **MapTag1**、 **MappTag2**。
3. 在客户端应用程序中，创建一个包含 **Pair1** 查询 （“热备份”对的名称）的 “分布式报警显示”对象。
4. 运行客户端应用程序，显示包含该 “分布式报警显示”对象的窗口。
5. 现在该 “分布式报警”对象将显示 Machine A 的报警。
6. 在主 “报警供应器” **MachineA** 上，确认某个报警，譬如说是 **Tag2**。
7. 此时辅助 “报警供应器” **MachineB** 上对应的已映射报警 **MapTag2** 将自动得到确认。

确认同步示例

假设在**选定关键码域**对话框中，**报警名**、**报警类**以及**报警类型**已被选作**设计时**关键码域，**报警组**已被选作**运行时**关键码域。



对应的报警记录 .csv 文件 (**Mapfile.csv**) 已使用 Microsoft Excel 创建。



	A	B	C	D	E	F
1	名称	分类	类型	名称	分类	类型
2	Tag1	VALUE	HiHi	MapTag1	VALUE	HiHi
3	Tag2	VALUE	Lo	MapTag2	VALUE	Lo

在通过**映射报警记录**对话框导入 **Mapfile.csv** 文件时，名称为 **tag1**、“类”为 **VALUE** 且 “类型”为 **HIHI** 的报警映射到名称为 **maptag1**、“类”为 **VALUE** 且 “类型”为 **HIHI** 的报警。因此，只要在 “主”报警节点确认 **Tag1** 的 **HiHi** 报警，该项确认也会出现在辅助 “报警供应器”节点与客户端节点中 **MapTag1** 的 **HiHi** 报警上，只要主节点的 **Tag1** 与辅助节点的 **MapTag1** 的**报警组**名 （运行时映射的字段）保持相同。

相反，如果在辅助 “报警供应器”节点上确认 **MapTag2** 的 **Lo** 报警，则客户端与主 “报警”节点上 **Tag2** 的 **Lo** 报警会自动得到确认，只要 **MapTag2** 与 **Tag2** 的 “报警组”名 （运行时映射的字段）保持相同。

仅当**设计时**和**运行时**映射匹配时，才会发生确认同步。

备注 您可以选择**设计时**与**运行时**报警记录字段的任意组合来进行映射。不过，切记要确保这些映射不会导致多重引用。

例如，如果在**选定关键码域**对话框中选定了两个**报警记录域**，譬如如**类**和**优先级**，则非常可能有一个以上的报警符合标准。在这种情况下，将无法保证 “热备份”同步能够正常工作。在传播确认的过程中，也可能会出现符合标准的随机报警，而其它匹配的报警可能会得不到确认。

分布式报警数据库视图

本章节介绍 InTouch 8.0 所提供的数据库视图。这些视图提供易于查询的虚拟表，供分析过去与当前发生的报警与事件。这些视图的主要目的在于，通过 Wonderware 长期 （包括后续的产品版本）支持的方式来分析报警与时间历史，而不管随着发行版的不断推出，实际的数据库架构发生了多么大的变化，存在多么大的差异。最后，提供视图还是为了与 AlarmSuite 产品保持向后兼容性。在安装产品并创建报警日志数据库之后，就可以使用所有的视图了。

视图与存储过程简介

InTouch “分布式报警系统”提供了一套工具，供您查看并分析由 Alarm DB Logger 记入关系型数据库的报警历史与事件信息。这些功能是以数据库视图与存储过程的形式提供的。

数据库视图可以看成是一个简单的逻辑表，通过一种简单易用的形式，将多个底层数据库表中的信息综合到一起。视图包含在数据库中。数据库视图就像是实际的表那样，可以使用标准的 SQL 进行查询，所以它们通常也被成为虚拟数据库表。查询视图时，会返回一组记录 （或称为行）。每行包含多个信息列，这些列包含着记录的数据。

分布式报警系统的数据库视图提供了一种查看自动化应用程序中所出现的报警历史与事件的方式。通过使用复杂的过滤标准对视图执行 SQL 命令，可以在这些视图上执行复杂的分析。例如，您可以检索工厂的特定区域在特定的时间段内发生的所有 HiHi 报警的报警记录。或者，您也可以检索某个特定报警供应器节点所记录的所有数据改变事件。

存储过程是 SQL 语句的集合，这些语句共同执行特定的功能并从数据库返回数据。因此，可以将它们看成是子程序或函数。存储过程可接受那些控制其如何操作的输入参数，然后以结果集的形式返回数据。存储过程的用户不必去使用 SQL 语法。相反，只要提供执行该过程所需的存储过程的名称与输入参数就可以了。返回的结果是一组记录，显示为 SQL 记录集，与视图非常相似。由于存储过程同它们内嵌的 SQL 语句存在于数据库中，减少了数据库与客户端之间的数据往返，因而还有助于提高系统性能。

InTouch "分布式报警系统"提供的存储过程可供用户从数据库提取信息，而不必了解 SQL 的细节。存储过程在内部承担大量的工作，如建立适当的 SQL 查询、连接并执行复杂的逻辑运算等。

存储过程与视图均可以让用户不必费心去了解数据库的内部细节，使得检索分析与报告报警和事件所需的信息非常容易，因此它们有着非常大的优势。

下面数节将定义随供的视图与存储过程，并通过几个简单易学的示例来展示它们使用之灵活及功能之强大。

视图与存储过程列的定义

存储过程与视图均会返回信息表。每个表包含许多列。本节介绍视图与存储过程 （稍后介绍）返回的所有列的定义。

列名	描述
AlarmCount	整型，表示指定时间段内报警发生的次数。如果报警在进入选定时间范围之前已经存在，则不会计算在内。
AlarmState	采用 Unicode 编码的字符串，表示报警的状态，如 UNACK_ALM （未确认 - 报警中）、 UNACK_RTN （未确认 - 恢复正常）、 ACK_ALM （已确认 - 报警中）、 ACK_RTN （已确认 - 恢复正常）。确认状态表示确认的同时已恢复正常。
AlarmType	详见 Type。
Area	采用 Unicode 编码的字符串，表示生成报警或事件的标记名或对象所属的报警组或区域的名称。
Category	采用 Unicode 编码的字符串，表示报警或事件的类或类别，如 Value、Dev、 ROC 等。
CheckValue	采用 Unicode 编码的字符串，表示报警的报警限值或事件的前一个值。
Comment	采用 Unicode 编码的字符串，表示报警的注释。
Description	采用 Unicode 编码的字符串，表示报警或生成报警的对象的描述字符串。如果是确认报警，则它是确认注释。
Domain Name	域的名称。
UNACKDuration	表示 "未确认持续时间"的毫秒数。对于 "未确认持续时间"，这表示最近的状态转换 （ALM 或子状态）与 ACK （如果有）之间经过的时间。对于 "报警持续时间"，这表示报警发生与返回正常之间经过的时间。
EngUnits	工程单位值 （将来）。 在 8.0 中返回值是 ""。

列名	描述
EventCount	在指定的时间段内发生事件的次数。
EventStamp	日期时间值，表示报警或事件的日期与时间，采用本地时间格式。
EventStampUTC (GMT)	日期时间值，表示报警或事件的日期与时间，采用通用协调时间格式。
GroupName	详见 Area。
LastEvent （将来）	日期时间值，表示此报警最近一次发生状态转换的日期与时间，采用本地时间格式。
Limit	采用 Unicode 编码的字符串，表示报警或事件发生时报警变量的报警限值。对于视图 v_AlarmSuiteAlarmLog，返回值为浮点型。
NodeName	采用 Unicode 编码的字符串，表示给分布式报警系统提供报警或事件的节点名。
Operator	采用 Unicode 编码的字符串，表示与报警或事件关联的操作员的用户名。
Priority	整型，表示报警优先值 (1-999)。
Provider	采用 Unicode 编码的字符串，表示给分布式报警系统提供报警或事件的应用程序或组件。
Tagname	采用 Unicode 编码的字符串，表示产生报警或事件的对象的名称。
TimeInState （将来）	整型，表示报警进入当前报警状态已经过的时间 （以秒计）。
Type	采用 Unicode 编码的字符串，表示类别中的报警类型，如 HiHi、 LoLo 等。
Units	工程单位字符串 （将来）。在 8.0 中的返回值是 ""。
User Full Name	操作员用户的全名 （如 Joseph P. Smith）。
Value	采用 Unicode 编码的字符串，表示报警或事件发生时变量的当前值。对于 视图 v_AlarmSuiteAlarmLog，返回值为浮点型。
ValueString	采用 Unicode 编码的字符串，表示报警或事件发生时变量的当前值。

要在 Enterprise Manager 中查看视图定义

1. 展开服务器组，然后展开某个服务器。
2. 展开**数据库**。
3. 展开 “视图”所属的数据库。
4. 单击**视图**。
5. 在详细资料窗格中，右击存储过程。
6. 单击**属性**

要调用某个视图

与存储过程不同，视图不带参数。要查看视图中的数据，请执行下列操作：

1. 在 Enterprise Manager 中，展开服务器组。

2. 展开服务器。
3. 展开**数据库**。
4. 展开 “视图” 所属的数据库。
5. 单击**视图**。
6. 在详细资料窗格中，右击 “视图”，然后选择**打开行**。
7. 单击**全部行**。

报警历史数据库视图

此视图提供选定时间范围内发生的所有报警与报警转换事件的历史列表。查询指定开始及结束日期与时间 （通过 EventStamp 或 EventStampUTC 列）。返回的记录包括报警来源、报警确认、报警启用、报警禁用，以及报警恢复正常等事件。请注意，视图中所有的字符串均采用 Unicode 编码方式。

v_AlarmHistory

列名	数据类型	描述
EventStamp	Datetime	报警事件的日期与时间 （数据库的本地时间）
AlarmState	nChar	报警状态：UNACK、 UNACK_RTN、 ACK、 ACK_RTN、 DISABLED （将来）、 ENABLED （将来）之一
Tagname	nChar	产生报警的对象的名称：如 TIC101
Description	nVarchar	报警的描述字符串。缺省值可以是对象描述 （或 InTouch 中的注释）。或者是，确认记录的确认注释。
Area	nChar	报警的 “区域” 或 “组” 的名称。
Type	nChar	报警的类型：如 Hi、 HiHi、 ROC、 PV.HiAlarm
Value	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值。
Priority	Integer	报警优先级。
Category	nChar	报警类或报警类别。如 Value、 Dev、 ROC、 Process、 Batch、 System 等。
Provider	nChar	报警的供应器：节点 /InTouch 或 GalaxyName。
Operator	nChar	操作员的用户名：如 JoeR （如果有）。
Domain Name	nChar	域的名称。
User Full Name	nChar	操作员用户的全名 （如 Joseph P. Smith）。
UNACK Duration	Float	这表示最近的状态转换 （ALM 或子状态）与 ACK （如果有）之间经过的时间。
User1	Float	用户自定义字段 1。
User2	Float	用户自定义字段 2。
User 3	nChar	用户自定义字段，字符串型。

列名	数据类型	描述
EventStampUTC	DateTime	报警事件的 UTC 日期 / 时间
Millisec	Small integer	事件标签秒值的分数部分，增量为 0.1 毫秒。

'SQL Query Analyzer' 中的 SQL 语句也可以写入，以查看 “视图”中的数据。例如，在 “查询”窗口中，输入：

```
Select top 100 * from v_AlarmHistory where Priority>10
   AND((Provider LIKE '%adelphi%')AND (Area LIKE '%$s%'))
```

-- 选择 v_AlarmHistory 视图中所有的记录

```
Select * from v_AlarmHistory
```

-- 选择 v_AlarmHistory 视图中优先级大于 100 的所有记录

```
Select * from v_AlarmHistory WHERE Priority >=100
```

辉亮显示 （选择）以上文本，然后按 **F5**。产生的数据将显示在窗口底部区域。

如需有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

v_AlarmHistory2

列名	数据类型	描述
EventStamp	Datetime	报警事件的日期与时间 （数据库的本地时间）
AlarmState	nChar	报警的状态：UNACK、 UNACK_RTN、 ACK、 ACK_RTN、 DISABLED （将来）、 ENABLED （将来）、 ACK_ALM、 UNACK_ALM 之一
Tagname	nChar	产生报警的对象的名称：如 TIC101
Description	nVarchar	报警的描述字符串。缺省值可以是对象描述 （或 InTouch 中的注释）。或者是，确认记录的确认注释。
Area	nChar	报警的 “区域”或 “组”的名称。
Type	nChar	报警的类型：如 Hi、 HiHi、 ROC、 PV.HiAlarm
Value	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值。
Priority	Integer	报警优先级。
Category	nChar	报警类或报警类别。如 Value、 Dev、 ROC、 Process、 Batch、 System 等。
Provider	nChar	报警的供应器：节点 /InTouch 或 GalaxyName。
Operator	nChar	操作员的用户名：如 JoeR （如果有）。
Domain Name	nChar	域的名称。
User Full Name	nChar	操作员用户的全名 （如 Joseph P. Smith）。
Alarm Duration	Float	报警发生与恢复正常之间经过的时间。
User1	Float	用户自定义字段 1。
User2	Float	用户自定义字段 2。
User 3	nChar	用户自定义字段，字符串型。

列名	数据类型	描述
EventStampUTC	DateTime	报警事件的 UTC 日期 / 时间
Millisec	Small integer	事件标签秒值的分数部分，增量为 0.1 毫秒。

'SQL Query Analyzer' 中的 SQL 语句也可以写入，以查看 “视图”中的数据。例如，在 “查询”窗口中，输入：

```
Select top 100 * from v_AlarmHistory2 where Priority>10
   AND((Provider LIKE '%adelphi%')AND (Area LIKE '%$s%'))
```

-- 选择 v_AlarmHistory2 视图中所有的记录

```
Select * from v_AlarmHistory2
```

-- 选择 v_AlarmHistory2 视图中优先级大于 100 的所有记录

```
Select * from v_AlarmHistory2 WHERE Priority >=100
```

辉亮显示 （选择）以上文本，然后按 **F5**。产生的数据将显示在窗口底部区域。

如需有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

事件历史数据库视图

此数据库视图提供选定的时间范围内发生的所有非报警事件的历史列表。查询客户端指定开始及结束日期与时间。返回的记录包括所有非报警事件。

v_EventHistory

列名	数据类型	描述
EventStamp	Datetime	事件的日期与时间。
Tagname	nChar	产生事件的对象的名称，如 Pump1。
Description	nVarChar	事件的描述字符串。缺省值可以是对象描述或 InTouch 中的注释。
Area	nChar	事件的 “区域”或 “组”的名称。
Type	nChar	事件的类型，如 “操作员数据改变”、“启动”等。
Value	nChar	新值 （如果有）。
CheckValue	nChar	旧值 （如果有）。
Category	nChar	事件类别或类，如 Value、Process、Batch、System 等。
Provider	nChar	事件的发生器，如节点 /InTouch 或用户更改的 “视图引擎”的名称。
Operator	nChar	操作员 1 的用户名：如 JoeR （如果有）。
Domain Name	nChar	域的名称。
User Full Name	nChar	操作员用户的全名 （如 Joseph P. Smith）。
User1	Float	用户自定义字段 1。
User2	Float	用户自定义字段 2。

列名	数据类型	描述
User 3	nChar	用户自定义字段，字符串型。
EventStampUTC	DateTime	事件的 UTC 日期 / 时间。
Millisec	Small integer	事件标签秒值的分数部分，增量为 0.1 毫秒。

报警与事件历史数据库视图

此数据库视图提供了选定的时间范围内发生的所有事件与报警的历史列表。查询客户端指定开始及结束日期与时间。返回的记录包括所有报警与事件。此视图将报警视图与事件视图合而为一，也就是会将这两个视图中的记录合并到一起。

v_AlarmEventHistory

列名	数据类型	描述
EventStamp	Datetime	事件的日期与时间。
AlarmState	nChar	报警的状态：UNACK、 UNACK_RTN、 ACK、 ACK_RTN、 DISABLED （将来）、 ENABLED （将来）之一。不适用于事件。
Tagname	nChar	产生报警的对象的名称，如 TIC101。
Description	nVarchar	报警 / 事件的描述字符串。缺省值可以是对象描述（或 InTouch 中的注释）。或者是，确认记录的确认注释。
Area	nChar	报警的 “区域”或 “组”的名称。
Type	nChar	报警或事件的类型，如 Hi、 HiHi、 ROC、 PV.HiAlarm、操作员数据改变等。
Value	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值，或事件的旧值。
Priority	Integer	报警优先级。
Category	nChar	报警或事件类，或报警类别，如 Value、 Process、 Batch、 System 等。
Provider	nChar	报警的供应器，如节点 /InTouch 或 GalaxyName。
Operator	nChar	确认操作员或数据改变操作员的用户名。
Domain Name	nChar	域的名称。
User Full Name	nChar	操作员用户的全名 （如 Joseph P. Smith）。
UNACK Duration	Float	最近报警状态发生转换到 “确认”之间经过的毫秒数。
User1	Float	用户自定义字段 1。
User2	Float	用户自定义字段 2。
User 3	nChar	用户自定义字段，字符串型。
EventStampUTC	DateTime	事件的 UTC 日期 / 时间
Millisec	Small integer	事件标签秒值的分数部分，增量为 0.1 毫秒。

'SQL Query Analyzer' 中的 SQL 语句也可以写入，以查看 “视图”中的数据。例如：

要：

从 v_AlarmEventHistory 视图中选择标记名为 "MyTag1"、报警状态为 ACK_RTN 或 ACK 的所有记录的 TagName、 Area 和 Type 列，并按 “供应器”排序

在 “查询”窗口中输入下面的语句：

```
Select TagName,Area,Type FROM v_AlarmEventHistory WHERE
    TagName='MyTag1' AND (AlarmState='ACK_RTN' OR
    AlarmState='ACK') ORDER BY Provider
```

辉亮显示 （选择）以上文本，然后按 **F5**。产生的数据将显示在窗口底部区域。如需有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

v_AlarmEventHistory2

列名	数据类型	描述
EventStamp	Datetime	事件的日期与时间。
AlarmState	nChar	报警的状态：UNACK、 UNACK_RTN、 ACK、 ACK_RTN、 DISABLED （将来）、 ENABLED （将来）之一。不适用于事件。
Tagname	nChar	产生报警的对象的名称，如 TIC101。
Description	nVarchar	报警 / 事件的描述字符串。缺省值可以是对象描述（或 InTouch 中的注释）。或者是，确认记录的确认注释。
Area	nChar	报警的 “区域”或 “组”的名称。
Type	nChar	报警或事件的类型，如 Hi、 HiHi、 ROC、 PV.HiAlarm、操作员数据改变等。
Value	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值，或事件的旧值。
Priority	Integer	报警优先级。
Category	nChar	报警或事件类，或报警类别，如 Value、 Process、 Batch、 System 等。
Provider	nChar	报警的供应器，如节点 /InTouch 或 GalaxyName。
Operator	nChar	确认操作员或数据改变操作员的用户名。
Domain Name	nChar	域的名称。
User Full Name	nChar	操作员用户的全名 （如 Joseph P. Smith）。
Alarm Duration	Float	从报警发生到返回正常状态所经过的毫秒数。
User1	Float	用户自定义字段 1。
User2	Float	用户自定义字段 2。
User 3	nChar	用户自定义字段，字符串型。
EventStampUTC	DateTime	事件的 UTC 日期 / 时间
Millisec	Small integer	事件标签秒值的分数部分，增量为 0.1 毫秒。

'SQL Query Analyzer' 中的 SQL 语句也可以写入，以查看 “视图”中的数据。例如：

要：

从 v_AlarmEventHistory 视图中选择标记名为 "MyTag1"、报警状态为 ACK_RTN 或 ACK 的所有记录的 TagName、 Area 和 Type 列，并按“供应器”排序

在 “查询”窗口中输入下面的语句：

```
Select TagName,Area,Type FROM v_AlarmEventHistory WHERE
    TagName='MyTag1' AND (AlarmState='ACK_RTN' OR
    AlarmState='ACK') ORDER BY Provider
```

辉亮显示 （选择）以上文本，然后按 **F5**。产生的数据将显示在窗口底部区域。如需有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

报警计数器数据库存储过程

此数据库存储过程提供指定的时间间隔内每种报警发生的次数。查询客户端必须指定开始及结束日期与时间。它包含五个可选参数：TagName、 Class、 Type、 Provider 以及 Comment。计数器仅适用于报警发生的次数 （不适用于确认或恢复正常等转换的次数）。因此，如果发生报警且随后它获得确认并返回正常状态，则该报警的计数仅仅是 1 （而不是 3）。此视图旨在显示报警频率。每个报警均可由其 TagName、Provider、 Type 以及 Category 唯一确定。此视图可以回答这样的问题：供应器 Node1|InTouch （供应器）上的对象 TIC101 （标记名）在某个时间段内共有多少次发出 HiHi （类型）的 “值”报警 （类别）？

sp_AlarmCounter

备注 报警计数器仅适用于详细模式的记录，合并模式的记录不支持它。

列名	数据类型	描述
Tagname	Nchar	产生报警的对象的名称，如 TIC101。
Area	Nchar	报警的 “区域”或 “组”的名称。
Type	Nchar	报警的类型，如 Hi、 HiHi、 ROC、 PV.HiAlarm。
Category	Nchar	报警类或报警类别，如 Value、 Process、 Batch 等。
AlarmCount	Integer	时间范围内报警发生的次数。如果报警在进入选定时间范围之前已经发生，则不会计算在内。
Priority	Integer	报警优先级。
Provider	Nchar	报警的供应器，如节点 /InTouch 或 GalaxyName。
Comment	Nchar	

在 Enterprise Manager 中查看存储过程的定义

要在 Enterprise Manager 中查看存储过程的定义

1. 展开服务器组，然后展开服务器。
2. 展开**数据库**。
3. 展开存储过程所属的数据库。
4. 单击**存储过程**。
5. 在详细资料窗格中，右击存储过程，然后单击**属性**。

调用存储过程

存储过程可以使用 Transact-SQL 语句 EXECUTE 从 SQL Server 调用。

下面是调用存储过程的一个示例：

在 'SQL Query Analyzer' 的 “查询”窗口中，输入：

```
EXECUTE sp_AlarmCounter @StartDate='2001-01-01',
   @EndDate='2001-03-31', @Tagname = 'tag1', @Type = 'LO',
   @Provider = 'WW21353\InTouch', @Comment = 'SSAADD'
```

辉亮显示 （选择）以上文本，然后按 **F5**。结果将显示在窗口底部区域。

StartDate 和 EndDate 是 sp_AlarmCounter 存储过程的两个必需提供的参数。其余五个参数属可选参数。这表示，您可以忽略除 StartDate 与 EndDate 之外的任何其它五个参数。忽略某个参数时，存储过程将不会使用它进行过滤，而显示包含所忽略参数的所有匹配值的一般结果。

StartDate 和 EndDate 是 SQL Server 中 'datetime' 类型的变量，可以接受 *SQL Server Users Guide* 中指定的各种格式的数据。您需要使用这两个参数来传递执行 “存储过程”所需的持续时间。

事件计数器数据库存储过程

此数据库存储过程提供某个时间间隔内在特定标记名上发生的特定类型的事件的次数。查询客户端必须指定开始及结束日期与时间。它包含三个可选参数：TagName、 Provider 及 Comment。此计数器仅适用于非报警事件。此视图旨在显示每种事件的发生频率。例如，泵打开了多少次？ TagnName、 Provider、 Category 及 Typ e 用于唯一确定事件并执行计数。

sp_EventCounter

列名	数据类型	描述
Tagname	NChar	产生报警的对象的名称，如 TIC101。
Area	NChar	报警的 “区域”或 “组”的名称。
Type	NChar	事件的类型。
Category	NChar	报警类或报警类别，如 Value、 Process、 Batch 等。

列名	数据类型	描述
EventCount	Integer	在指定的时间范围内标记名发生了此类型的事件的次数。
Provider	NChar	事件的供应器：节点 /InTouch 或 GalaxyName。
Comment	NChar	

与此类似， sp_EventCounter 存储过程也采用两个必须提供的参数 StartDate 与 EndDate，以及三个可选参数 Tagname、 Provider 及 Comment。

```
EXECUTE sp_AlarmCounter @StartDate='2001-01-01 23:23:23',
   @EndDate='2001-03-31 23:23:23', @Tagname = '$NewAlarm'
```

AlarmSuite 报警日志数据库视图

此数据库视图提供历史报警与事件的视图，它返回的表与实际的 Alarm Suite 表 AlarmLog 具有相同的列。在 AlarmSuite 的 AlarmLog 表上可以正常工作的查询在本视图上也可以正常工作，只不过这些查询中的表名必须更改为本视图的名称。

v_AlarmSuiteAlarmLog

列名	数据类型	描述
EventStamp	Datetime	事件的日期与时间。
EventType	NChar	由 AlarmSuite 定义的事件的类型。
AlarmType	NChar	报警的类型。
AlarmState	NChar	报警的确认状态。
NodeName	NChar	报警的节点。
Tagname	NChar	报告报警的对象的名称。
GroupName	NChar	报警组或区域的名称。
Comment	NChar	确认注释 （如果有）。
Value	Float	报警变量的值
Limit	Float	报警时报警限的值。
ValueString	NChar	详见 Alarm Suite。
Operator	NChar	操作员姓名。
Priority	Integer	优先级
Units	NChar	在 7.11 中返回值是 ""。

'SQL Query Analyzer' 中的 SQL 语句也可以写入，以查看 “视图”中的数据。例如，在 “查询”窗口中输入：

```
-- 从 v_AlarmSuiteAlarmLog 视图选择所有不同的标记名
```

Select Distinct TagName From v_AlarmSuiteAlarmLog

辉亮显示 （选择）以上文本，然后按 F5。产生的数据将显示在窗口底部区域。

如需有关 SQL 语句的详细信息，请参阅 SQL Server 手册。

第　１０　章

报警 / 事件客户端

InTouch 提供一个 Alarm Viewer ActiveX 控件，它具有内置滚动条、可调大小的显示列、可选择多个报警、更新状态栏、动态显示类型以及基于报警优先级的显示颜色等功能。Alarm Viewer ActiveX 控件是查看报警的首选控件。不过， InTouch 还包含了一个分布式报警显示，可加速查看现有报警的过程，而不必重新配置它们以便配合使用 AlarmViewer ActiveX 控件。

InTouch 提供了一个 Alarm DB View ActiveX 控件，可供您以可视化方式查看 Alarm DB Logger 数据库中的报警数据。此控件可用于查看所有的报警和事件信息。

InTouch 还提供了 InTouch QuickScript 函数，供您动态控制报警显示与报警确认。下面将针对每个控件介绍 QuickScript 的用法。

目录

Alarm Viewer ActiveX 控件指南

InTouch 提供一个 Alarm Viewer ActiveX 控件，供您查看本地和远程生成的所有报警。通过指定下列属性，您可以设计 ActiveX 控件和显示数据的外观。

- 上下文相关菜单功能
- 显示模式
- 列表控制选项
- 不同属性的颜色
- 字体类型、样式和大小

- 报警选择 （过滤器）
- 查询过滤器
- 列管理
- 排序

一旦设计控件格式，用户将能够进行以下调整来操纵其所查看的数据：

- 按列对信息排序
- 更新显示
- 执行查询
- 调整列宽

Alarm Viewer ActiveX 控件可放入 WindowMaker，并可以调整大小和位置。配置 Alarm Viewer ActiveX 控件的工作可以通过使用 “属性”页来完成。随后，用户便可以在 Alarm Viewer ActiveX 控件的视图窗口中查看报警数据。

Alarm Viewer ActiveX 控件指南

Alarm Viewer ActiveX 显示是一个附加的报警查看器控件向导。虽然它看上去有些象旧的标准显示向导，但它使用的机制却与 “窗口控制”向导相同。此机制要求在使用 Alarm Viewer ActveX 显示之类的对象时，须遵循一些准则。这些准则如下：

每个显示都必须有一个标识符，以便关联的 QuickScript 函数知道需要修改哪个显示。此标识符是在 **Alarm Viewer ActiveX 属性**对话框的**控件名**中输入的，对于每个显示，它必须保持唯一。

显示不应覆盖其它的 InTouch 对象，如窗口控件或图形对象等。通过在 WindowMaker 中单击 Alarm Viewer ActiveX 显示，并检查该显示 “手柄”，可以很方便地验证这点。这些手柄不应碰到屏幕上其它的图形对象。

显示不应随意使用。在屏幕上放置大量显示，会导致系统性能下降。如有可能，应限制屏幕上的显示数量，必要时请调用含其它显示的其他屏幕。

安装 Alarm Viewer ActiveX 控件

此 Alarm Viewer ActiveX 控件随 InTouch 一起安装。

要在 WindowMaker 窗口中粘贴此控件

1. 打开 “向导选择”对话框。
2. 选择 ActiveX 控件下的 **AlmViewerCtrl**，然后单击**确定**。
3. 将控件粘贴到窗口上，并调整到所需的大小。

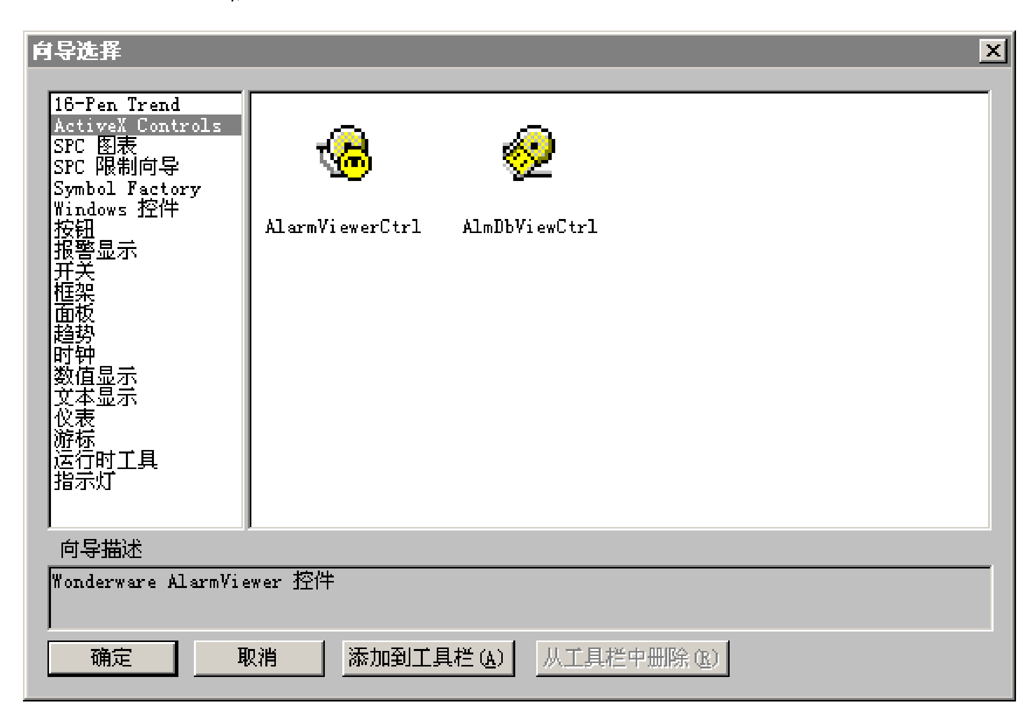
卸载 Alarm Viewer ActiveX 控件

1. 删除粘贴到窗口上的所有 Alarm Viewer 控件。
2. 从**特别**菜单上，选择 “配置”。
3. 选择**向导 /ActiveX 安装**，然后打开 “向导 /ActiveX 安装”对话框。
4. 选择 **ActiveX 控件**安装属性页。此时 Wonderware Alarm Viewer 控件名将显示在 “安装的 ActiveX 控件”文本区。
5. 单击 Wonderware Alarm Viewer 控件，然后单击**删除**。
6. 在显示警告消息时单击**是**。
7. 单击**关闭**。

创建 Alarm Viewer ActiveX 控件

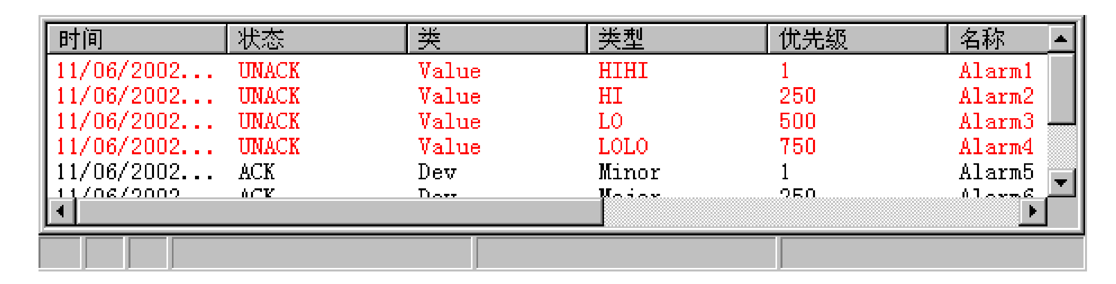
要创建 Alarm Viewer ActiveX 显示

1. 单击**向导 /ActiveX** 工具栏中的**向导**工具。此时会出现**向导选择**对话框。



2. 从向导列表中选择 **ActiveX 控件**。
3. 双击 **AlarmViewerCtrl** 控件，或选择它，然后再单击**确定**。此时对话框关闭，您的窗口会再次出现，且光标处于*粘贴*模式。

4. 在窗口中单击以粘贴 Alarm Viewer 控件。



提示 若要调整向导大小，请指向它的一个选择手柄，然后拖拽它直至达到所需大小。

---

5. 现在，您便可以开始配置显示。

访问 AlmViewerCtrl 属性对话框

您可以通过下列方法来访问 “AlmViewerCtrl 属性”对话框：

- 双击控件，或
- 右击控件，然后从弹出式菜单中选择 “属性”菜单。

配置 Alarm Viewer ActiveX 控件

AlarmViewerCtrl 对话框有 7 个属性页。这些属性页分别是**控件名**、**通用**、**颜色**、**时间格式**、**查询**、**属性**及**事件**。

备注 配置对话框与所有的标准 Windows 属性页相似，在单击**确定**或**应用**之前不会记录任何设置。不过，您从一个属性页 （选项卡）切换到另一个时，程序会去验证适当项目的选项是否有效。如果某个项目验证不成功，则会重新回到包含该项目的属性页，并且会显示一个消息框，指出所发生的错误。当您从一个属性页切换到另一个时，会自动应用对前一个属性页所作的更改。如果单击**取消**，则会忽略所有输入并关闭对话框。

Alarm Viewer ActiveX 显示属性

要配置 Alarm Viewer ActiveX 显示

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrl** 对话框，并显示**控件名**属性页。

提示 如果使用右键单击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的命令。

---

2. 单击要配置的属性页的选项卡。

配置控件名和显示位置

要配置控件名和显示位置

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrl** 对话框，并显示**控件名**属性页。



2. 在**控件名**框中，输入报警显示的名称。对于所用的每个报警显示，此名称均必须保持唯一。

备注 在缺省条件下，“控件名”由该控件的 ProgID 确定。 ProgID 是在计算机中安装 ActiveX 控件时输入系统注册表的名称。将该控件的实例放入 InTouch 应用程序时，程序会从系统注册表读取该控件的 ProgID，并在其后附加一个索引号，结果就生成了控件名，如 **AlarmViewerCtrl1**。

提示 此处输入的名称将在整个系统中用以引用此对象，以执行确认报警与查询之类的任务。

3. 在**左侧**、**宽度**、**顶部**及**高度**等对话框中输入数值，可以调整该控件的报警查看器位置：

选项	描述
左侧	确定 Alarm Viewer 显示与窗口左边缘的距离。位置数值越低，显示越靠近窗口的左边缘。位置数值越高，显示越远离窗口的左边缘。
宽度	确定 Alarm Viewer 显示的宽度。
顶部	确定 Alarm Viewer 显示与窗口顶部边缘的距离。位置数值越低，显示越靠近窗口的顶部边缘。位置数值越高，显示越远离窗口的顶部边缘。
高度	确定 Alarm Viewer 显示的高度。
可见的	取消选择**可见的**复选框可使得控件在运行期间不可见。
GUID	显示此 ActiveX 控件的唯一标识。

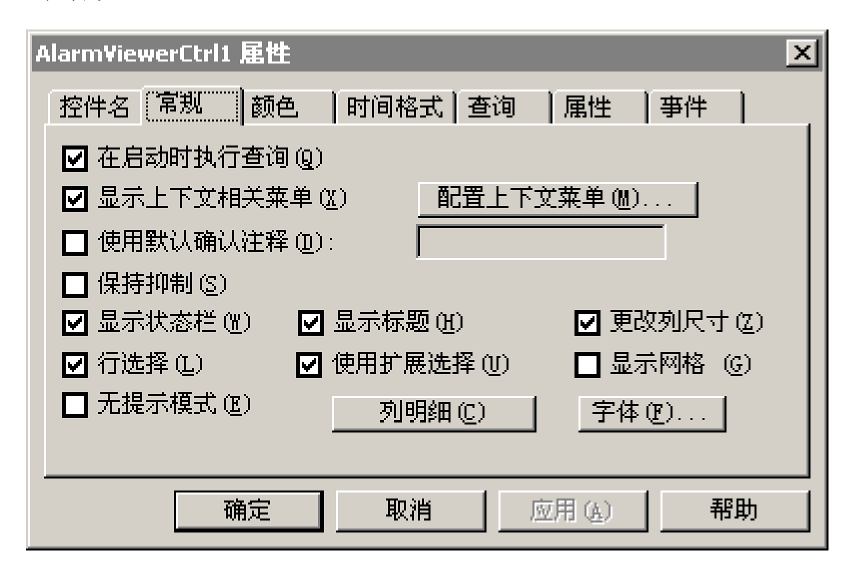
4. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

配置通用属性

要配置通用属性

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。

2. 单击**通用**选项卡以激活**通用**属性页。此时**通用**属性页会显示选定的缺省设置。



3. 单击各个复选框，以选择通用显示选项。取消选择复选框，可以在 Alarm Viewer 中忽略这些选项。下表将介绍这些复选框选项 :

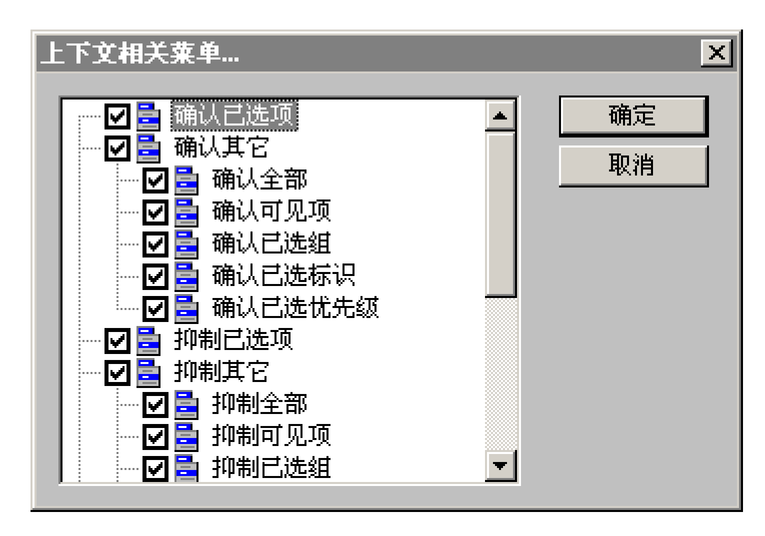
属性	描述
启动时执行查询	如果选定此复选框，则会使用缺省查询属性自动开始更新显示。如果未选定，您需要在显示更新以前，执行 **almDefQuery** 或 **almQuery**。**重新查询**选项也可以从右击快捷菜单中选取。
显示上下文相关菜单	激活右击弹出式菜单的功能。选择此选择可启用**配置上下文菜单**按钮和**使用默认确认注释**复选框。
使用默认确认注释	控制在操作员确认报警时是否使用缺省注释。如果选定了此复选框，并输入了一个字符串，则该字符串将被用作运行期间的缺省注释。如果未选定此复选框，则在操作员确认报警时，会显示一个对话框，提醒操作员输入一则注释。您既可以填写该对话框，也可以将其留为空白。选择此选项时，会启用 “确认注释”字段，供用户输入注释。只有在选定**显示上下文相关菜单**复选框时，此按钮才可用。
显示状态栏	切换状态栏的显示状况。
显示标题	切换标题栏的显示状况。
行选择	选择此选项时，可供您在 Alarm Viewer 中选择记录。此选项允许用户选择多个报警，并且不会根据新选择的报警而取消选择先前选择的报警。
使用扩展选择	允许通过按住 Ctrl 或 Shift 键并配合使用鼠标选择多项报警。缺省情况是通过单击它们来切换选择的报警 （只有在选定**行选择**复选框时才可用）。
无提示模式	如果选定了**无提示模式**，则分布式报警显示不会在运行时弹出任何错误消息。如果未选定它，则报警显示会弹出错误消息。错误消息总是会发送给 Log Viewer。
更改列尺寸	如果选定了**更改列尺寸**，则用户可以在运行时调整列的大小。如果未选定它，则用户无法在运行时更改列的大小。
显示网格	如果选定了**显示网格**，则 Alarm Viewer 会在分布式报警显示中显示网格。如果未选定，则网格是不可见的。

4. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

配置上下文相关菜单

要配置上下文相关菜单

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**通用**选项卡以激活**通用**属性页。
3. 单击**配置上下文菜单**按钮，以选择要在运行时显示于上下文相关鼠标右击菜单中的选项。此时会出现 “上下文相关菜单”屏幕。



备注 只有在选定**显示上下文相关菜单**选项时**配置上下文菜单**按钮才会被启用。

4. 配置上下文相关菜单选项，然后单击**确定**。如果选定了**显示上下文相关菜单**复选框，则至少应选择一个上下文相关菜单项。
5. 下面介绍各个上下文相关菜单选项：

菜单选项	描述
确认所选项	供用户去确认选定的报警。
确认其它项	供用户去按照其它方法确认报警。用户可以从下面非粗体的选项中选择要确认哪些报警。如果选定了**确认其它项**，则您至少要选择一个子菜单项。
确认全部	供用户去确认所有活动的报警。
确认可见项	供用户去确认可见的报警。
确认已选组	供用户去确认使用相同供应器名的选定组。
确认已选标识	供用户去确认使用相同供应器、组和优先级的选定标记。
确认已选优先级	供用户去确认使用相同供应器、组和优先级的选定优先级。
抑制所选项	供用户去抑制选定的报警。

菜单选项	描述
抑制其它项	供用户去按照其它方法抑制报警。用户可以从下面非粗体的选项中选择要抑制哪些报警。如果选定了**抑制其它项**，则您至少要选择一个子菜单项。
抑制全部	供用户去抑制所有的报警。
抑制可见项	供用户去抑制可见的报警。
抑制已选组	供用户去抑制选定的组。
抑制已选标识	供用户去抑制选定的标记。
抑制已选优先级	供用户去抑制选定的优先级。
全部撤销抑制	供用户去撤销抑制的所有报警。
查询收藏夹	供用户访问“查询收藏夹”对话框。
统计	供用户访问并查看报警统计。在运行时右击 Alarm Viewer 显示，然后单击快捷菜单中的**统计**。此时会显示**报警统计**对话框，在对话框的标题中指出这些报警统计资料来自哪个控件。
抑制	供用户去访问“抑制”对话框。
冻结	供用户在冻结 / 撤销冻结模式之间切换显示。
重新查询	供用户去重新运行报警查询。
排序	供用户去访问“排序”对话框。

备注 如果设计时**上下文相关菜单**对话框中的**确认所选项**和**确认其它项**这两个菜单项均未选定，则会禁用**使用默认确认注释**复选框和文本框。单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

配置显示列明细

要配置显示列明细

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**通用**选项卡以激活**通用**属性页。

3. 单击**列明细**按钮。此时会出现“列明细”对话框。

列明细

名称	宽度	原名
时间	100	时间
状态	100	状态
类	100	类
类型	100	类型
优先级	100	优先级
名称	100	名称
组	100	组
供应器	100	供应器
值	100	值
报警限	100	报警限

移动

确定 (O)
取消 (C)
编辑 (E)...
重置为默认值 (R)

4. 从**列明细**对话框中，选择**列名**旁的复选框，以便在报警对象中显示该列。**列明细**对话框中的列有**名称**、**宽度**及**原名**。**原名**显示在发生更改之前，这些列的名称是什么。下面介绍您可以选择显示原始的列名。

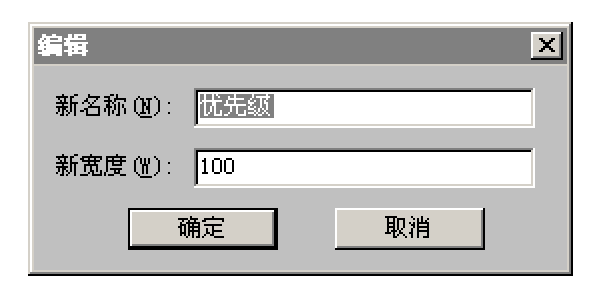
备注 您至少应选择一个列。

列名	描述
时间	按“时间格式”属性页中选定的格式显示时间。
状态	显示报警状态。
类别	显示报警类别。
类型	显示报警类型。
优先级	显示报警优先级。
名称	显示报警 / 标记名。
组	显示“报警组”名称。
供应器	显示报警供应器的名称。
值	显示报警发生时标记名的值。列宽应该足够大以满足精度要求。
报警限	显示标记名的报警限值。列宽应该足够大以满足精度要求。
操作员	显示与报警条件关联的已登录操作员的 ID。
注释	显示标记名的注释。这些注释是在数据库中定义标记名时输入“报警注释”框的。给报警引入“确认注释”时，新的注释会在这个“注释”列中更新。
操作员用户名	显示与报警关联的操作员的用户名。
操作员域	显示与报警关联的操作员所在的域。

列名	描述
节点名	显示与报警关联的节点名。
标记注释	显示与报警关联的标记注释。

备注 缺省情况下，除**操作员**与**注释**外，所有列名均被选择。

5. 若要重新整理各个列，请选择列名，然后使用 "上移" 与 "下移" 箭头键。**列明细**对话框顶部出现的列名即为在报警显示最左边显示的列。
6. 若要编辑列名和宽度，请双击列名或选择一个列名，然后单击**编辑**。此时会出现该列的**编辑**对话框。



7. 如果您要显示与缺省列名不同的列名，请在**新名称**文本框中输入新的名称。
8. 在**新宽度**文本框中输入列宽。列宽按像素计算，范围是从 1 至 999 像素。缺省列宽为 100 像素。
9. 单击**编辑**对话框上的**确定**。

备注 单击**重置为默认值**以使之回到缺省的**列明细**设置。

10. 单击**列明细**对话框上的**确定**。
11. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

配置 ActiveX Alarm Viewer 字体属性

要配置字体属性

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。

2. 单击**通用**选项卡，然后单击**字体**按钮。此时会显示 “字体”对话框：

3. 滚动 “字体”列表，然后选择所需的字体类型。
4. 滚动**字体样式**列表，以选择一种字体样式。
5. 滚动**大小**列表，以选择一种字体大小。
6. 在**效果**区，选中**删除线**复选框或**下划线**复选框，以选择 “删除线”或 “下划线”属性。
7. 单击**字符集**下列箭头以选择所需的脚本类型。

备注 **样本**框会显示所选字体属性的示例。

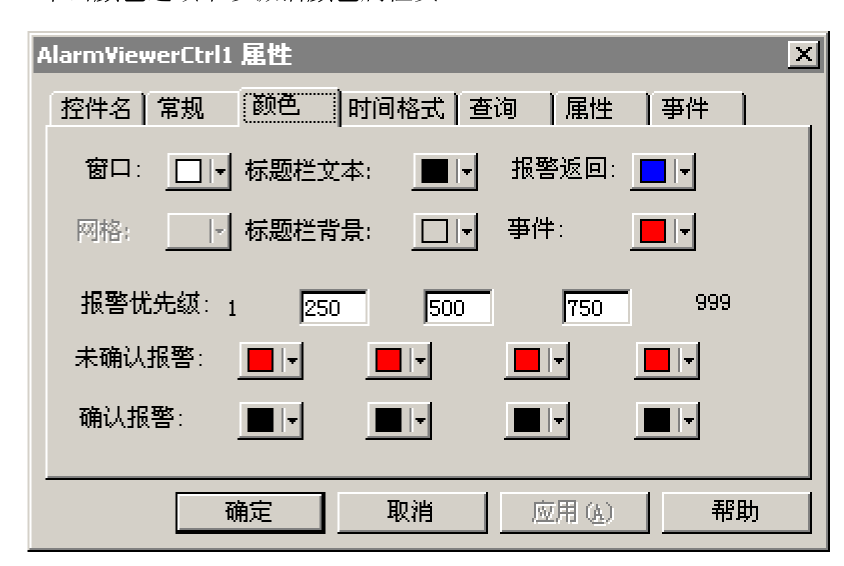
8. 单击**确定**。

配置 ActiveX Alarm Viewer 颜色属性

要配置报警显示颜色

1. 双击 Alarm Viewer ActiveX，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。

2. 单击**颜色**选项卡以激活**颜色**属性页。



3. 单击每个颜色框以打开调色板。在调色板上为下列各项单击您要使用的颜色：

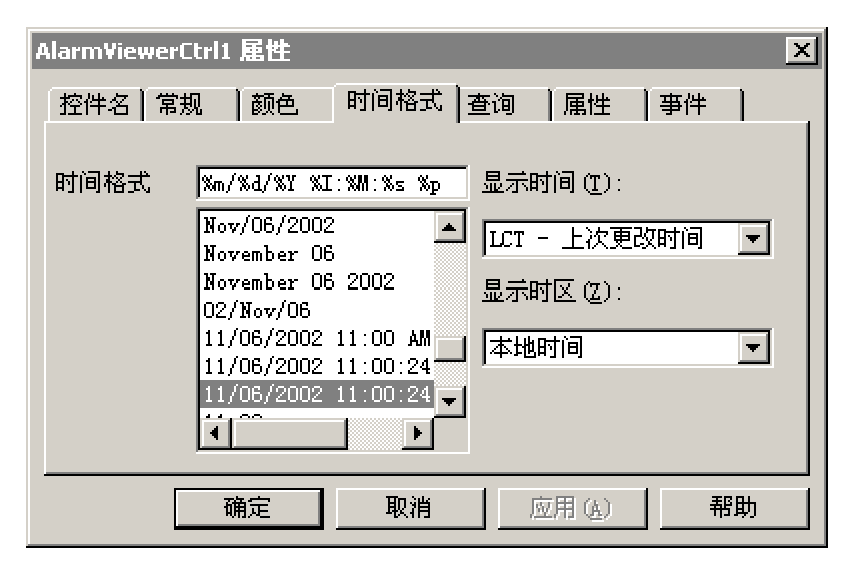
属性	描述
窗口	设置显示背景颜色。
标题栏文本	设置标题栏文本颜色（仅当选定 “显示标题”选项时此项才可用）。
报警返回	设置返回的报警颜色（未经确认返回到正常状态的报警）。
网格	设置网格颜色。缺省情况下不会显示网格。缺省网格颜色是淡灰色。报警对象中网格的颜色自动设置为与所选**窗口**颜色的对比色。
标题栏背景	设置标题栏背景颜色（仅当选定 “显示标题”选项时此项才可用）。
事件	设置事件报警颜色。

4. 在**报警优先级**框中，输入用于报警显示的断点值。
5. 单击**未确认报警**与**确认报警**颜色框以打开调色板。单击调色板中您要使用的颜色。
6. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

配置 **Alarm Viewer** 时间格式

要配置 Alarm Viewer 时间格式

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**时间格式**选项卡以激活**时间格式**属性页。



滚动 “时间格式”选项，以选择所需的时间格式，或配置附加的时间格式选项。时间格式字符串由 % 号分隔的字符串组成。下表介绍时间格式字符串字符：

字符串字符	描述
d	两位数字的日期 - 09
b	3 个字母的月份缩写 - Aug
Y	四位数字的年份 - 2002
m	两位数字的月份 - 08
y	两位数字的年份 - 02
#x	完整的日期与星期 - Friday, August 09, 2002
B	完整的月份名称 - August
H	24 小时制表示的时间 - 16:00
M	分钟 00:41
P	PM
S	秒钟 - 16:41:07
s	小数秒 - 16:41:07.390
I	要求指定 AM/PM 的 12 小时制时间表示法 - 04:41 PM

下面是一些时间格式字符串示例：

时间格式字符串	显示
%d %b	09 Aug
%m/%d/%Y	08/09/2002
%#x	Friday, August 09, 2002
%Y-%m-%d	2002-08-09
%m/%d/%Y %H:%M %p	08/09/2002 16:56 PM
%m/%d/%Y %H:%M:%s %p	08/09/2002 16:56:38.07 PM
%I:%M %p	04:56 PM

3. 单击**显示的时间**下拉箭头以选择显示的时间。可用的选项包括：

OAT	原始报警时间 - 即报警发生的日期 / 时间标签。
LCT	上次更改时间 - 即报警实例最近一次改变状态的日期 / 时间标签：包括报警发生、子状态改变、恢复正常或确认。
LCT，但确认后用 OAT	“上次更改时间”，但确认后用 “原始报警时间” - 也就是，未确认报警时使用 LCT，但在确认报警后使用 OAT。

4. 单击**显示时区**下拉箭头以选择时区。可用的选项包括：

GMT	格林尼治标准时间，也称作 “通用协调时间”、 UTC 或 Zulu。
本地时间	报警时间已调整为本地时区。
原始时间	报警时间已调整为报警源的时区。

5. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

配置显示报警查询

要配置报警显示查询

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**查询**选项卡以激活**查询**属性页。



3. 选择所需的查询选项。下表介绍这些查询选项：

属性	描述
从优先级	缺省情况下，最小报警优先级设置为 1。此值总应小于 “到优先级”，并且取值范围是从 1 到 999。
到优先级	缺省情况下，最大报警优先级设置为 999。此值总应大于 “从优先级”，并且取值范围是从 1 到 999。
报警状态	缺省情况下，要查询的报警状态会设置为 “全部”。 “全部”表示查询所有的报警。 “未确认”表示查询未确认的报警。 “确认”表示查询已确认的报警。

属性	描述
查询类型	将显示类型设置为“摘要”或“历史”。
报警查询	设置初始的报警查询。此字段只接受文本；它不接受标记。缺省“报警查询”是 **\intouch!$system** 这些列表的有效语法包括： **\\Node\InTouch!Group** “报警组”的完整路径 **\InTouch!Group** 本地“报警组”的完整路径 **GroupList** 另一个组列表
排序列	使用下拉箭头设置要排序的列。选项包括报警“时间”、“状态”、“类别”、“类型”、“优先级”、“名称”、“组”、“供应器”、“值”、“报警限”、“操作员”以及“注释”。
查询收藏夹文件	控件会存储报警查询。单击按钮可浏览查询收藏夹。“查询收藏夹”与“报警查询”之间的区别有： ï “报警查询”文件是 XML。 ï 文件名应包括正确的文件夹名，以便在运行时启用“查询收藏夹”上下文相关菜单选项。 ï 现有的文件名允许访问现有的查询。创建新查询时，会创建新的文件名。 ï 查询文件可以在任意文件夹中，并且不必在应用程序文件夹下。
自动滚动到新报警	只有在选定基于“时间”的排序选项时，此复选框才可以使用。
排序方向	选择给查询排序的标准，单击“向上”或“向下”单选钮可分别按升序或降序进行排序。缺省情况下，显示按时间的升序进行排序。

备注 若要执行多项查询，请用空格将每项查询分开。

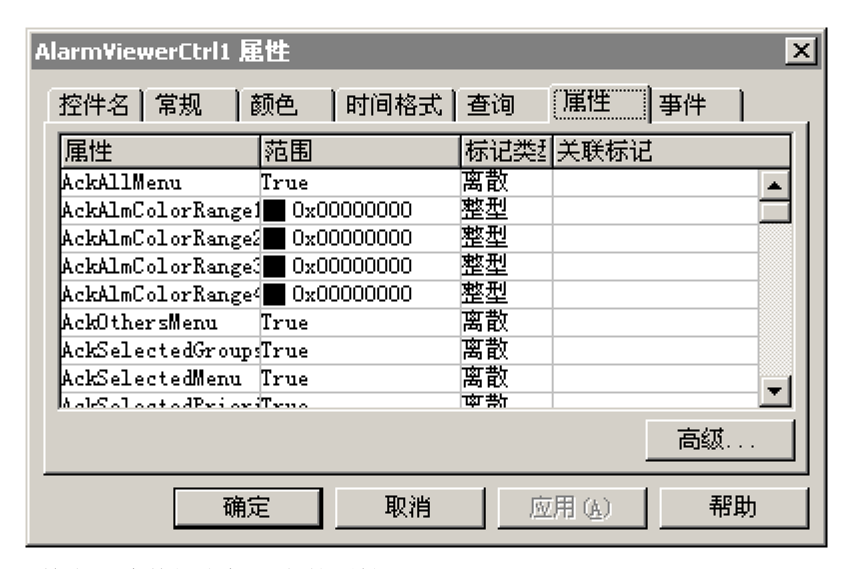
例如：**\\Master\InTouch!MyGroup LocalGroupList**

4. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

将标记名指定给控件属性

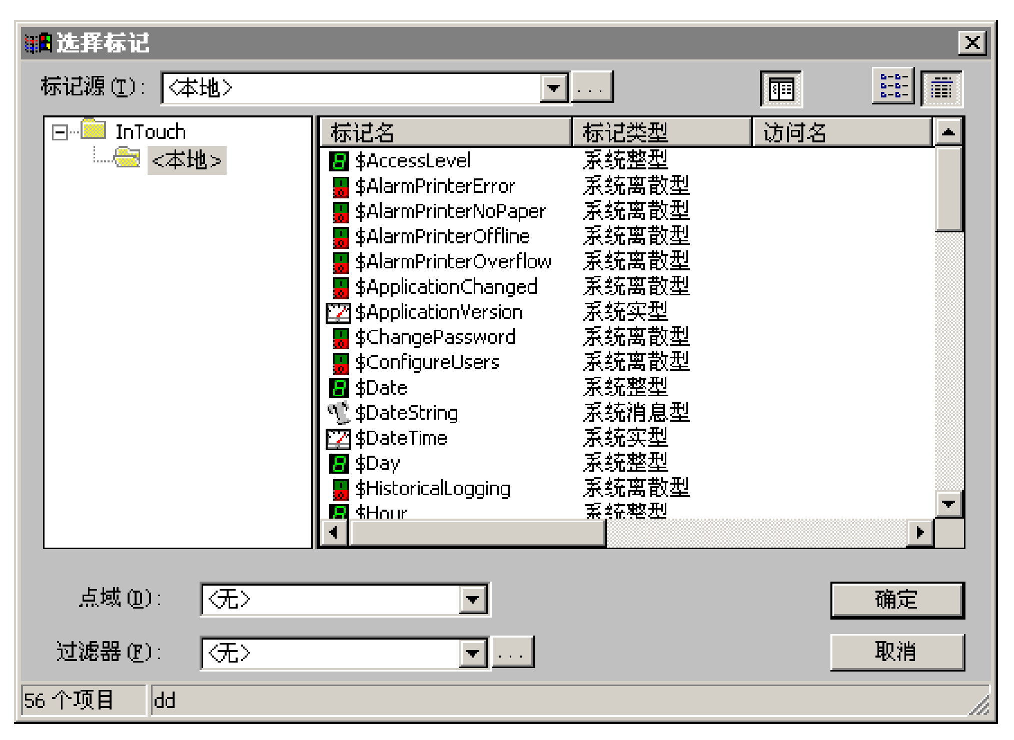
要将标记名指定给控件属性

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**属性**选项卡以激活**属性**属性页。



3. 单击要给其指定标记名的属性。

4. 单击 “关联标记” 列中的按钮以浏览标记名。此时会出现**选择标记**窗口。



5. 单击**确定**。
6. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

AlarmViewer 属性选项卡属性

若要设置属性，请输入 #object.PropertyName = 1; 或 #object.PropertyName = tag1; 其中， object 是 AlarmViewer 的名称， tag1 是一个离散型标记。例如，若要设置 AckAllMenu 属性，请输入 #AlarmViewer.AckAllMenu = 1。

若要获取属性，请输入 tag1 = #object.PropertyName; 其中， object 是 AlarmViewer 的名称， tag1 是一个离散型标记。例如，若要获取 AckAllMenu 属性，请输入 tag1 = #AlarmViewer.AckAllMenu;。

InTouch 可以接受英文双引号括起的单词 "True" 和 "False"，并将它们分别当作值 1 和 0。例如，动作脚本：

#AlarmViewerCtrl2.SortMenu = "False";

会从 Alarm Viewer ActiveX 控件的右击菜单中删除 “排序菜单”。

属性选项卡上的属性如下：

属性：	**AckAllMenu**
用途：	布尔值属性。启用 / 禁用 “确认全部”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckAllMenu *[= discrete]*

属性：	**AckAlmColorRange1**
用途：	设置用于显示优先级范围在 1 到 ColorPriorityRange1 之间已确认报警的颜色。缺省优先级范围是 1 到 250。
类型：	整型
缺省值：	黑色
语法：	*Object*.AckAlmColorRange1 *[= Integer]*

属性：	**AckAlmColorRange2**
用途：	设置用于显示优先级范围在 ColorPriorityRange1 到 ColorPriorityRange2 之间已确认报警的颜色。缺省优先级范围是 250 到 500。
类型：	整型
缺省值：	黑色
语法：	*Object*.AckAlmColorRange2 *[= Integer]*

属性：	**AckAlmColorRange3**
用途：	设置用于显示优先级范围在 ColorPriorityRange2 到 ColorPriorityRange3 之间已确认报警的颜色。缺省优先级范围是 500 到 750。
类型：	整型
缺省值：	黑色
语法：	*Object*.AckAlmColorRange3 *[= Integer]*

属性：	**AckAlmColorRange4**
用途：	设置用于显示优先级范围在 ColorPriorityRange3 到 999 之间已确认报警的颜色。缺省优先级范围是 750 到 999。
类型：	整型
缺省值：	黑色
语法：	*Object*.AckAlmColorRange4 *[= Integer]*

属性：	**AckOthersMenu**
用途：	布尔值属性。启用 / 禁用 “确认其他”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckOthersMenu *[= discrete]*

属性：	**AckSelectedGroupsMenu**
用途：	布尔值属性。启用 / 禁用 “确认已选组”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckSelectedGroupsMenu*[= discrete]*

属性：	**AckSelectedMenu**
用途：	布尔值属性。启用 / 禁用 “确认已选项”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckSelectedMenu *[= discrete]*

属性：	**AckSelectedPriorities**
用途：	布尔值属性。启用 / 禁用 “确认已选优先级”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckSelectedPriorities *[= discrete]*

属性：	**AckSelectedTags**
用途：	布尔值属性。启用 / 禁用 “确认已选标记”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckSelectedTags *[= discrete]*

属性：	**AckVisibleMenu**
用途：	布尔值属性。启用 / 禁用 “确认可见项”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.AckVisibleMenu *[= discrete]*

属性：	**AlarmQuery**
用途：	设置初始的报警查询。此字段只接受文本；它不接受标记。这些列表的有效语法包括： **\\Node\InTouch!Group** “报警组”的完整路径 **\InTouch!Group** 本地 “报警组”的完整路径 **GroupList** 另一个组列表
类型：	消息
缺省值：	\intouch!$system
语法：	*Object*.AlarmQuery *[= message]*

属性：	**AlarmState**
用途：	缺省报警状态查询 （“全部”、“未确认”、“确认”）。
类型：	消息
缺省值：	全部
语法：	*Object*.AlarmState *[= message]*

属性：	**AlmRtnColor**
用途：	设置报警返回颜色 （报警已返回到正常状态，而没有被确认），不管是“已确认”还是 “未确认”。
类型：	整型
缺省值：	全部
语法：	*Object*.AlmRtnColor *[= Integer]*

属性：	**AutoScroll**
用途：	如果用户从开始处滚动列表，则这会自动跳到新报警。（新报警定义为目前尚未显示在显示对象中的那些报警）。
类型：	离散
缺省值：	False
语法：	*Object*.Autoscroll *[= discrete]*

属性：	**ColorPriorityRange1**
用途：	设置要显示的报警的优先级范围之边界。此属性的值必须大于 1 且小于 ColorPriorityRange2。
类型：	整型
缺省值：	250
语法：	*Object*.ColorPriorityRange1 *[= 整数或优先级]*

属性：	**ColorPriorityRange2**
用途：	设置要显示的报警的优先级范围之边界。此属性的值必须大于 ColorPriorityRange1 且小于 ColorPriorityRange3。
类型：	整型
缺省值：	500
语法：	*Object*.ColorPriorityRange2 *[= 整数或优先级]*

属性：	**ColorPriorityRange3**
用途：	设置要显示的报警的优先级范围之边界。此属性的值必须大于 ColorPriorityRange2 且小于 999。
类型：	整型
缺省值：	750
语法：	*Object*.ColorPriorityRange3 *[= 整数或优先级]*

属性：	**ColumnResize**
用途：	返回或设置一个用于决定是否可在运行时调整列大小的值。
类型：	离散
缺省值：	True
语法：	*Object*.ColumnResize *[= Discrete]*

属性：	**DefaultAckComment**
用途：	字符串属性。在 "UseDefaultAckComment" 为 TRUE 的情况下确认报警时，这会被用作注释，否则会提示用户输入注释。
类型：	消息
缺省值：	无
语法：	*Object*.DefaultAckComment *[= message]*

属性：	**DisplayedTime**
用途：	显示报警消息时间。
类型：	消息
缺省值：	LCT - 上次更改时间
语法：	*Object*.DisplayedTime *[= message]*

属性：	**DisplayedTimeZone**
用途：	字符串属性。返回 / 设置当前时区字符串。值只能是 "GMT"、“原始时间”或 “本地时间”。
类型：	消息
缺省值：	本地时间
语法：	*Object*.DisplayedTimeZone *[= message]*

属性：	**EventColor**
用途：	设置事件报警颜色。
类型：	整型
缺省值：	红色
语法：	*Object* EventColor *[= color]*

属性：	**ExtendedSelection**
用途：	允许通过按住 Ctrl 或 Shift 键并配合使用鼠标选择多项报警。缺省情况是通过单击它们来切换选择的报警 （只有在选定**行选择**复选框时才可用）。
类型：	离散
缺省值：	True
语法：	*Object*.ExtendedSelection *[= discrete]*

属性：	**Font**
用途：	设置在控件中显示记录与标题所用的字体。
类型：	无
缺省值：	Times New Roman
语法：	*只读*

属性：	**FreezeMenu**
用途：	布尔值属性。启用 / 禁用 “冻结”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.FreezeMenu *[= discrete]*

属性：	**FromPriority**
用途：	设置缺省查询的最低优先级值。
类型：	整型
缺省值：	1
语法：	*Object*.FromPriority *[= integer]*

属性：	**GridColor**
用途：	设置背景网格的颜色。
类型：	整型
缺省值：	灰色
语法：	*Object*.GridColor *[= color]*

属性：	**QueryFavoritesFile**
用途：	字符串属性。返回 / 设置查询收藏夹文件名。
类型：	消息
缺省值：	无
语法：	*Object*.QueryFavoritesFile *[= message]*

属性：	**QueryFavoritesMenu**
用途：	启用 / 禁用 "QueryFavorites" 菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.QueryFavoritesMenu *[= discrete]*

属性：	**QueryName**
用途：	返回当前查询名称。
类型：	字符串 （只读）
缺省值：	None
语法：	*Object*.QueryName *[= String]*

属性：	**QueryStartup**
用途：	使用缺省查询属性自动开始更新显示，如果已选定。如果未选定，您需要在显示更新以前，执行 ApplyDefaultQuery 或 ApplyQuery。
类型：	离散
缺省值：	True
语法：	*Object*.QueryStartup *[= discrete]*

属性：	**QueryType**
用途：	将显示类型设置为 “摘要” 或 “历史”。
类型：	消息
缺省值：	摘要
语法：	*Object*.QueryType *[= message]*

属性：	**RequeryMenu**
用途：	启用 / 禁用 “重新查询” 菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.RequeryMenu *[= discrete]*

属性：	**RetainSuppression**
用途：	更改报警查询时在报警查询之间保持报警抑制。
类型：	离散
缺省值：	True
语法：	*Object*.RetainSuppression *[= discrete]*

属性：	**RowSelection**
用途：	允许用户在运行期间选择报警。
类型：	离散
缺省值：	True
语法：	*Object*.RowSelection *[= discrete]*

属性：	**SelectedCount**
用途：	返回选定报警的总数。
类型：	整型 （只读）
缺省值：	None
语法：	*Object*.SelectedCount *[= integer]*

属性：	**ShowContextMenu**
用途：	激活右击弹出式菜单的功能。
类型：	离散
缺省值：	True
语法：	*Object*.ShowContextMenu *[= discrete]*

属性：	**ShowGrid**
用途：	返回或设置一个值，该值确定是否在控件中显示网格线。
类型：	离散
缺省值：	False
语法：	*Object*.ShowGrid *[= discrete]*

属性：	**ShowHeading**
用途：	显示控件的标题栏。
类型：	离散
缺省值：	True
语法：	*Object*.ShowHeading *[= discrete]*

属性：	**ShowStatusBar**
用途：	返回或设置一个值，该值确定是否显示状态栏。状态栏包含三个指示器：状态消息、当前报警查询及进度条。这些指示器提供了关于显示查询当前状态的概貌，并可提供关于 “分布式报警对象”中可用抑制的详细信息。冻结生效时，状态栏的右窗格呈红色；抑制生效时，状态栏的左窗格呈红色。抑制生效时，左侧窗格会显示 “抑制”字样。
类型：	离散
缺省值：	True
语法：	*Object*.ShowStatusBar *[= discrete]*

属性：	**SilentMode**
用途：	返回或设置一个值，该值确定控件是否处于无提示模式。
类型：	离散
缺省值：	False
语法：	*Object*.SilentMode *[= discrete]*

属性：	**SortColumn**
用途：	字符串属性。返回 / 设置当前排序列。
类型：	消息
缺省值：	时间
语法：	*Object*.SortColumn *[= message]*

属性：	**SortOrder**
用途：	字符串属性。返回 / 设置排序方向。可能的值有 “升序” 与 “降序”，分别用 0 与 1 表示。
类型：	离散
缺省值：	False
语法：	*Object*.SortOrder *[= discrete]*

属性：	**SortMenu**
用途：	布尔值属性。启用 / 禁用 “排序” 菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SortMenu *[= discrete]*

属性：	**StatsMenu**
用途：	布尔值属性。启用 / 禁用 “统计” 菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.StatsMenu *[= discrete]*

属性：	**SuppressAllMenu**
用途：	布尔值属性。启用 / 禁用 “抑制全部”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressAllMenu *[= discrete]*

属性：	**SuppressedAlarms**
用途：	返回抑制报警的总数。
类型：	整型 （只读）
缺省值：	None
语法：	*Object*.SuppressedAlarms *[= integer]*

属性：	**SuppressionMenu**
用途：	布尔值属性。启用 / 禁用 “抑制”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressionMenu *[= discrete]*

属性：	**SuppressOthersMenu**
用途：	布尔值属性。启用 / 禁用 “抑制其他”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressOthersMenu *[= discrete]*

属性：	**SuppressSelectedGroupsMenu**
用途：	布尔值属性。启用 / 禁用 “抑制已选组”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressSelectedGroupsMenu *[= discrete]*

属性：	**SuppressSelectedMenu**
用途：	布尔值属性。启用 / 禁用 “抑制已选项”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressSelectedMenu *[= discrete]*

属性：	**SuppressSelectedPriority**
用途：	布尔值属性。启用 / 禁用 “抑制已选优先级”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressSelectedPriority *[= discrete]*

属性：	**SuppressSelectedTagsMenu**
用途：	布尔值属性。启用 / 禁用 “抑制已选标记”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressSelectedTagsMenu *[= discrete]*

属性：	**SuppressVisibleMenu**
用途：	布尔值属性。启用 / 禁用 “抑制可见项”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.SuppressVisibleMenu *[= discrete]*

属性：	**TimeFormat**
用途：	设置报警显示时间格式。
类型：	消息
缺省值：	%b%y （如 27 Aug）
语法：	*Object*.TimeFormat *[= message]*

属性：	**TitleBackColor**
用途：	设置标题栏背景颜色 （仅当选定 “显示标题”选项时此项才可用）。
类型：	整型
缺省值：	灰色
语法：	*Object*.TitleBackColor *[= color]*

属性：	**TitleForeColor**
用途：	设置标题栏前景颜色 （仅当选定 “显示标题”选项时此项才可用）。
类型：	整型
缺省值：	黑色
语法：	*Object*.TitleForeColor *[= color]*

属性：	**ToPriority**
用途：	设置报警查询的最大优先级。
类型：	整型
缺省值：	999
语法：	*Object*.ToPriority *[= integer]*

属性：	**TotalAlarms**
用途：	返回报警的总数。
类型：	整型 （只读）
缺省值：	None
语法：	*Object*.TotalAlarms *[= integer]*

属性：	**UnackAlarms**
用途：	返回未确认报警的总数。
类型：	整型 （只读）
缺省值：	None
语法：	*Object*.UnackAlarms *[= integer]*

属性：	**UnAckAlmColorRange1**
用途：	根据值范围设置 “未确认” 报警的颜色。此属性的值必须大于 1 且小于 UnAckAlmColorRange2。
类型：	整型
缺省值：	红色
语法：	*Object*.UnAckAlmColorRange1 *[= color]*

属性：	**UnAckAlmColorRange2**
用途：	设置用于显示优先级范围在 ColorPriorityRange1 到 ColorPriorityRange2 之间未确认报警的颜色。
类型：	整型
缺省值：	红色
语法：	*Object*.UnAckAlmColorRange2 *[= color]*

属性：	**UnAckAlmColorRange2**
用途：	设置用于显示优先级范围在 ColorPriorityRange2 到 ColorPriorityRange3 之间未确认报警的颜色。
类型：	整型
缺省值：	红色
语法：	*Object*.UnAckAlmColorRange3 *[= color]*

属性：	**UnAckAlmColorRange3**
用途：	设置用于显示优先级范围在 ColorPriorityRange3 到 999 之间未确认报警的颜色。
类型：	整型
缺省值：	红色
语法：	*Object*.UnAckAlmColorRange4 *[= color]*

属性：	**UnsuppressAllMenu**
用途：	布尔值属性。启用 / 禁用 “全部撤销抑制”菜单项。
类型：	离散
缺省值：	True
语法：	*Object*.UnsuppressAllMenu *[= discrete]*

属性：	**UseDefaultAckComment**
用途：	字符串属性。如果 "UseDefaultAckComment" 为 TRUE，则在确认报警时，会使用默认的 “确认注释”。否则会提示用户输入注释。
类型：	离散
缺省值：	False
语法：	*Object*.UseDefaultAckComment *[= discrete]*

属性：	**WindowColor**
用途：	设置显示背景颜色。
类型：	整型
缺省值：	白色
语法：	*Object*.WindowColor *[= color]*

方法与事件

方法：	**AboutBox**
用途：	显示 “关于” 对话框。
语法：	*Object*.AboutBox
示例：	`#AlarmViewerCtrl1.AboutBox();` （其中，控件的名称是 AlarmViewerCtrl1）。

方法：	**AckSelected**
用途：	Alarm Viewer 允许在运行时通过单击鼠标来选择报警。**AckSelected** 函数可用于确认这些报警。
语法：	*Object*.AckSelected
示例：	`#AlarmViewerCtrl1.AckSelected();`

方法：	**AckAll**
用途：	确认当前报警查询中的所有报警。由于 Alarm Viewer 的显示区域有限，所以 **almAckAll** 函数可能会确认显示中不可见的报警。
语法：	*Object*.AckAll
示例：	`#AlarmViewerCtrl1.AckAll();`

方法：	**AckVisible**
用途：	仅确认 Alarm Viewer 中当前可见的那些报警。
语法：	*Object*.AckVisible
示例：	`#AlarmViewerCtrl1.AckVisible();`

方法：	**AckSelectedGroup**
用途：	确认与一个或多个所选的报警具有相同组名且来自相同供应器的所有报警。
语法：	*Object*.AckSelectedGroup
示例：	`#AlarmViewerCtrl1.AckSelectedGroup();`

方法：	**AckSelectedTag**
用途：	确认来自相同供应器与组名、使用相同的标记名且与一个或多个所选的报警具有相同优先级的所有报警。
语法：	*Object*.AckSelectedTag
示例：	`#AlarmViewerCtrl1.AckSelectedTag();`

方法：	**AckSelectedPriority**
用途：	确认来自相同供应器与组名且与一个或多个所选的报警具有相同优先级的所有报警。
语法：	*Object*.AckSelectedPriority
示例：	`#AlarmViewerCtrl1.AckSelectedPriority();`

方法：	**ShowSuppression**
用途：	显示抑制对话框，该对话框包含所有抑制的报警。
语法：	*Object*.ShowSuppression
示例：	`#AlarmViewerCtrl1.ShowSuppression();`

方法：	**SuppressSelected**
用途：	抑制显示所选报警的当前与将来的实例。
语法：	*Object*.SuppressSelected
示例：	`#AlarmViewerCtrl1.SuppressSelected();`

方法：	**SuppressAll**
用途：	抑制显示所有活动报警的当前与将来的实例。
语法：	*Object*.SuppressAll
示例：	`#AlarmViewerCtrl1.SuppressAll();`

方法：	**SuppressVisible**
用途：	抑制显示任何可见报警的当前与将来的实例。
语法：	*Object*.SuppressVisible
示例：	`#AlarmViewerCtrl1.SuppressVisible();`

方法：	**SuppressSelectedGroup**
用途：	抑制显示与一个或多个具有相同 “供应器” 名的所选报警属于相同报警组的任何报警的当前与将来的实例。
语法：	*Object*.SuppressSelectedGroup
示例：	`#AlarmViewerCtrl1.SuppressSelectedGroup();`

方法：	**SuppressSelectedTag**
用途：	抑制显示与具有相同 “供应器” 名、“组” 名及 “优先级” 范围的一个或多个所选报警使用相同 “标记名” 的任何报警的当前与将来的实例。
语法：	*Object*.SuppressSelectedTag
示例：	`#AlarmViewerCtrl1.SuppressSelectedTag();`

方法：	**SuppressSelectedPriority**
用途：	抑制显示与一个或多个具有相同 “供应器” 名与 “组” 标记的所选报警具有相同优先级的当前与将来的显示。
语法：	*Object*.SuppressSelectedPriority
示例：	`#AlarmViewerCtrl1.SuppressSelectedPriority();`

方法：	**UnSuppressAll**
用途：	撤销报警抑制。
语法：	*Object*.UnSuppressAll
示例：	`#AlarmViewerCtrl1.UnSuppressAll();`

方法：	**ShowQueryFavorites**
用途：	显示 “查询收藏夹” 对话框。
语法：	*Object*.ShowQueryFavorites
示例：	`#AlarmViewerCtrl1.ShowQueryFavorites();`

方法：	**ShowStatistics**
用途：	显示 “统计” 对话框。
语法：	*Object*.ShowStatistics
示例：	
	`#AlarmViewerCtrl1.ShowStatistics();`

方法：	**FreezeDisplay**
用途：	冻结显示。
语法：	*Object*.FreezeDisplay
示例：	`#AlarmViewerCtrl1.FreezeDisplay();`

方法：	**Requery**
用途：	再次查询报警供应器。
语法：	*Object*.Requery
示例：	`#AlarmViewerCtrl1.Requery();`

方法：	**AckGroup**
用途：	确认来自相同报警供应器且具有给定组名的所有报警。
语法：	*Object*.AckGroup
示例：	`#AlarmViewerCtrl1.AckGroup();`

方法：	**AckPriority**
用途：	确认具有相同报警供应器名与组名且优先级在指定范围内的所有报警。
语法：	*Object*.AckPriority
示例：	`#AlarmViewerCtrl1.AckPriority();`

方法：	**AckTag**
用途：	确认在给定优先级范围内具有相同 “标记名”、“供应器”名及 “组”名的所有报警。
语法：	*Object*.AckTag
示例：	`#AlarmViewerCtrl1.AckTag();`

方法：	**SuppressGroup**
用途：	抑制显示属于给定组名且使用相同 “供应器”名的任何报警当前与将来的实例。
语法：	*Object*.SuppressGroup
示例：	`#AlarmViewerCtrl1.SuppressGroup();`

方法：	**SuppressPriority**
用途：	抑制显示具有相同 “供应器”名与 “组”名且优先级在指定范围内的任何报警当前与将来的显示。
语法：	*Object*.SuppressPriority
示例：	`#AlarmViewerCtrl1.SuppressPriority();`

方法：	**SuppressTag**
用途：	抑制显示通过给定标记名来确定的所有报警的当前及未来的实例，这些报警具有相同的 “供应器名”、“组名” 及 “优先级” 范围。
语法：	*Object*.SuppressTag
示例：	`#AlarmViewerCtrl1.SuppressTag();`

方法：	**SelectGroup**
用途：	切换选择给定 “供应器” 中来自给定 “报警组” 的所有报警。
语法：	*Object*.SelectGroup
示例：	`#AlarmViewerCtrl1.SelectGroup();`

方法：	**SelectPriority**
用途：	切换选择给定 “供应器”、“报警组” 内优先级在给定范围内的所有报警。
语法：	*Object*.SelectPriority
示例：	`#AlarmViewerCtrl1.SelectPriority();`

方法：	**SelectTag**
用途：	切换选择特定报警 “供应器 / 组 / 标记” 内的所有报警。您也可以指定 “优先级” 范围，或使用 1-999。
语法：	*Object*.SelectTag
示例：	`#AlarmViewerCtrl1.SelectTag();`

方法：	**ApplyQuery**
用途：	执行摘要或历史报警信息查询。此函数中提供了所有的查询属性。
语法：	*Object*.ApplyQuery
示例：	`#AlarmViewerCtrl1.ApplyQuery();`

方法：	**ApplyDefaultQuery**
用途：	使用 “配置”对话框的缺省属性执行查询。这些属性包括：“从优先级”、“到优先级”、“报警列表”以及 “显示类型”。缺省属性只能在开发时更改，并且不会被其它报警查询改写。
语法：	*Object*.ApplyDefaultQuery
示例：	`#AlarmViewerCtrl1.ApplyDefaultQuery();`

方法：	**SelectAll**
用途：	切换选择显示中的所有报警。由于报警显示的显示区域有限，所以 **SelectAll** 函数可能会选择显示中不可见的报警。
语法：	*Object*.SelectAll
示例：	`#AlarmViewerCtrl1.SelectAll();`

方法：	**SelectItem**
用途：	选择 *nItem* 处的报警记录。
语法：	*Object*.SelectItem
示例：	`#AlarmViewerCtrl1.SelectItem();`

方法：	**UnSelectAll**
用途：	取消选择所有选定的记录。
语法：	*Object*.UnSelectAll
示例：	`#AlarmViewerCtrl1.UnSelectAll();`

方法：	**SetQueryByName**
用途：	使用与用户定义的查询名关联的查询参数开始一项新的报警查询。
语法：	*Object*.SetQueryByName
示例：	`#AlarmViewerCtrl1.SetQueryByName();`

方法：	**MoveWindow**
用途：	提供用于控制显示窗口的命令。这些命令包括：向上翻页 (Page Up)、向下翻页 (Page Down)、向右滚动 (Scroll Right)、向左滚动 (Scroll Left)、向上一行 (Line Up)、向下一行 (Line Down)、顶部 (Top)、底部 (End) 等。
语法：	*Object*.MoveWindow
示例：	`#AlarmViewerCtrl1.MoveWindow();`

方法：	**GetItem**	
用途：	返回指定行和列的数据为一字符串。	
语法：	*Object*.GetItem*(Integer, message)*	
	整型	表示控件中特定某行的整型表达式。
	消息	表示控件中的列名的字符串表达式。
示例：	`tagname = #AlarmViewerCtrl1.GetItem(1, "Group");`	

方法：	**ShowContext**
用途：	在启用 "RefreshMenu" 或 "ResetMenu" 或 "SortMenu" 等任一属性的情况下，显示上下文相关菜单。
语法：	*Object*.ShowContext
示例：	`#AlarmViewerCtrl1.ShowContext();`

方法：	**ShowSort**
用途：	显示 “辅助排序” 对话框。
语法：	*Object*.ShowSort
示例：	`#AlarmViewerCtrl1.ShowSort();`

方法：	**SetSort**
用途：	设置通过 "SortColumn" 与 "SortOrder" 属性来指定的排序标准。
语法：	*Object*.SetSort
示例：	`#AlarmViewerCtrl1.SetSort();`

将 ActiveX 脚本指定给 Alarm Viewer

要将 ActiveX 脚本指定给 Alarm Viewer

1. 双击 Alarm Viewer ActiveX 显示，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**事件**选项卡以激活**事件**属性页。



3. 单击要给其指定脚本的事件。
4. 在 “脚本” 字段中输入脚本名，或单击按钮来浏览 ActiveX 脚本。如果您选择浏览 ActiveX 脚本，请在选定脚本后单击 “确定”。
5. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

事件：	**Click**	
用途：	用户在控件上单击并释放鼠标左键后发生。	
语法：	*tagname = #Thisevent*.clicknRow	
	nRow	整型表达式，反映发生的事件所在的行。
	#Thisevent	关键字

备注 从 OnStartup 事件调用此方法时， ActiveX Alarm Viewer 控件会忽略 UI 方法，因为此控件尚不是可见的。下面列出了一些 UI 方法，如果它们是从 OnStartup 事件中调用的，则会被忽略掉："ShowSort"、 "ShowContext"、 "GetSelectedItem"、 "GetNext"、 "GetPrevious" 以及 "AboutBox"。

事件：	**DoubleClick**	
用途：	用户在控件上双击并释放鼠标左键时发生。	
语法：	*Tagname = #ThisEvent*.DoubleClicknRow	
	NRow	整型表达式，反映发生的事件所在的行。
	#Thisevent	关键字

Click 与 DoubleClick 事件均是零基的。给用户发布 Click 与 / 或 DoubleClick 事件时，显示中的行计数是从 0 开始的

快捷菜单

ActiveX Alarm Viewer 对象提供了一个快捷菜单，供您快速访问命令，以应用于运行时屏幕中包含的显示对象和 （或）一个或多个所选报警、“报警组”、“标记名”以及 “优先级”。

- **确认所选项** - 确认选定的报警。
- **确认其它项** - 操作员右击屏幕并指向**确认其它项**时，可打开一个包含确认命令的子菜单。
  - **全部确认** - 确认当前报警查询中所有的报警。由于报警显示的显示区域有限，所以 **AckAll** 函数可能会确认显示中不可见的报警。
  - **确认可见项** - 仅确认在报警显示中当前可见的报警。
  - **确认已选组** - 确认与一个或多个所选的报警具有相同组名且来自相同供应器的所有报警。
  - **确认已选标识** - 确认来自相同供应器与组名、使用相同的标记名且与一个或多个所选的报警具有相同优先级的所有报警。
  - **确认已选优先级** - 确认来自相同供应器与组名且与一个或多个所选的报警具有相同优先级的所有报警。

备注 在已经确认的报警上进行 “确认”没有任何效果。

- **抑制所选项** - 运行时操作员右击显示并指向**抑制所选项**时，可快速抑制所选的报警。
- **抑制其它项** - 操作员右击显示并指向**抑制其它项**时，可打开一个包含抑制命令的子菜单。
    - **抑制全部** - 抑制显示所有报警的当前与将来的实例。
    - **抑制可见项** - 抑制显示任何可见报警的当前与将来的实例。
    - **抑制已选组** - 抑制显示与一个或多个具有相同 “供应器”名的所选报警属于相同报警组的任何报警的当前与将来的实例。
    - **抑制已选标识** - 抑制显示特定的所有报警的当前及未来实例，这些报警与一个或多个选定的报警具有相同的 “标记名”名称，这些选定的报警具有相同的 “供应器名”、“组名”及 “优先级”范围。
    - **抑制已选组** - 抑制显示与一个或多个具有相同 “供应器”名的所选报警属于相同报警组的任何报警的当前与将来的实例。
    - **全部撤销抑制** - 清除抑制设置。
- **查询收藏夹** - 右击显示并指向**查询收藏夹**时，会出现一个**报警查询**对话框。

  如需有关详细信息，请参阅 “选择与配置报警查询收藏夹”。
- **统计** - 打开 “报警统计”对话框。
- **抑制** - 打开 “报警抑制”对话框。
- **冻结** - 冻结当前显示。
- **重新查询** - 再次查询报警供应器。
- **抑制** - 打开 “辅助排序”对话框。

分布式报警显示

“分布式报警系统”使用单个显示对象来显示本地与远程产生的报警。此显示对象的功能包括：内置滚动条、可调显示列、多项报警选择、更新状态栏、查询状态栏、上下文相关快捷菜单，以及基于报警优先级的报警显示颜色。

InTouch 允许您修改报警显示的外观 （包括所显示的信息）、各种报警状态所用的颜色、“报警组”，以及显示的报警优先级。

日期	时间	状态	类	类型	优先级	名称
06 十一月	11:37	UNACK	Value	HIHI	1	Alarm1
06 十一月	11:37	UNACK	Value	HI	250	Alarm2
06 十一月	11:37	UNACK	Value	LO	500	Alarm3
06 十一月	11:37	UNACK	Value	LOLO	750	Alarm4
06 十一月	11:37	ACK	Dev	Minor	1	Alarm5
06 十一月	11:37	ACK	Dev	Major	250	Alarm6
06 十一月	11:37	ACK	ROC	1	500	Alarm7

滚动条

分布式报警显示具有内置的水平与垂直滚动条，可供您滚动查看所列的各项报警。您可以配置这些滚动条的显示方式。

下一页 / 上一页控件

分布式报警显示也支持 **.NextPage** 与 **.PrevPage** 报警显示控制属性。

如需有关详细信息，请参阅 “报警显示控制属性”。

可调显示列

分布式报警显示使用网格来放置报警消息。此网格允许动态调整列宽，您只要选择一个列并拖动它，直至得到所需列宽即可。此功能只在运行时可用。您可以配置能否使用网格来调整列的大小。

提示 网格列的更改不会得以保存；因此，如果您在更改网格列后关闭包含报警显示的窗口，再次打开窗口时，网格列又会使用缺省的宽度。双击垂直网格线可自动调整列的大小。

多项选择

网格允许您在一个列表框中选择一个或多个报警。所选报警可通过使用 **almAckSelect()** QuickScript 函数进行确认，本章稍后部分将介绍此函数。在配置分布式报警显示时，您还可以定义选择动作，以允许切换单项选择 （逐项）或多项选择 （在按下 CTRL 或 SHIFT 键的同时，通过单击鼠标选择多项报警）。您可以关闭运行时选择。

报警消息颜色

根据报警的优先级以及是否已确认，您还可以为每个显示的报警消息配置最多八种不同的颜色。

更新状态栏

分布式报警显示包括一个状态栏，该状态栏含有三个指示器：状态消息、当前报警查询及进度条。这些指示器提供了关于显示查询当前状态的概貌，并可提供关于 “分布式报警对象”中可用抑制的详细信息。冻结生效时，状态栏的右窗格呈红色；抑制生效时，状态栏的左窗格呈红色。抑制生效时，左侧窗格会显示 “抑制”字样。在运行期间，您可以关闭状态栏的显示。

更新成功 | 缺省查询

功能	描述
状态消息	状态栏左端的状态消息提供关于当前查询状态的更详尽说明。
进度条	状态栏右端的更新进度条提供关于当前查询进度的形象化指示。
报警查询	“报警查询”提供当前报警查询的形象化指示。 **备注** 如果在冻结显示时收到新的报警，状态栏中会显示 “新报警”字样。

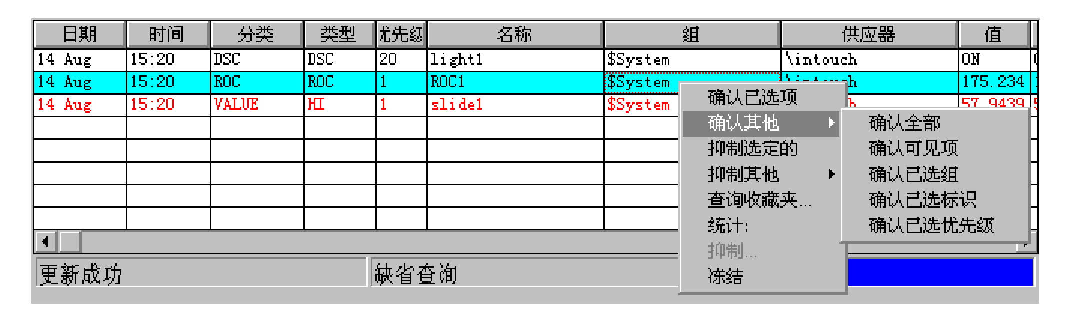
状态消息	状态 / 指示器	进度条
无	无查询	无
未完全更新	查询未完成	蓝 / 绿
更新成功	查询结束	实心蓝色
抑制	查询名	实心蓝色
冻结	查询名	红色
新报警	查询名	红色

快捷菜单

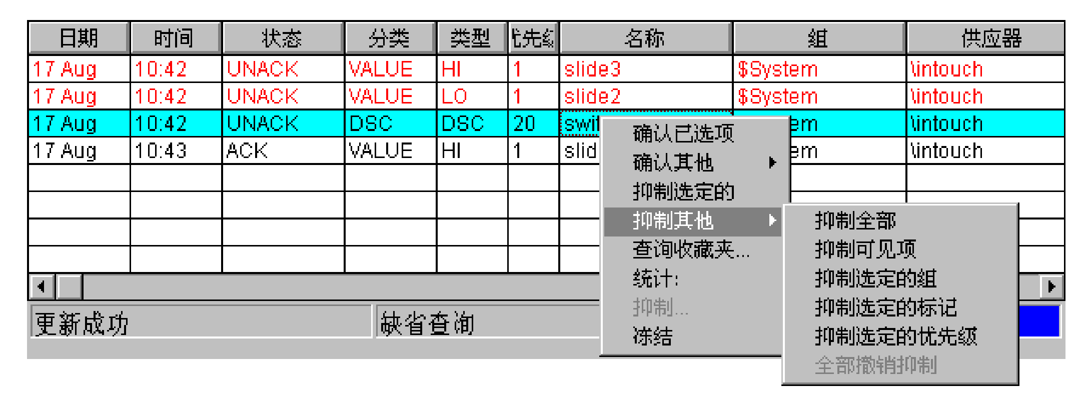
“分布式报警显示”对象提供了一个快捷菜单，供您快速访问命令，以应用于运行时屏幕中包含的显示对象和 （或）一个或多个所选报警、“报警组”、“标记名”以及 “优先级”。

备注 缺省条件下，快捷菜单是禁用的。通过在 “属性”页中设置相应的值，可以启用它。此外，使用快捷菜单还可调用脚本函数来执行任何所需的操作。

- **确认所选项** - 运行时操作员右击显示并指向**确认所选项**时，可快速确认所选的报警。
- **确认其它项** - 操作员右击屏幕并指向**确认其它项**时，可打开一个包含确认命令的子菜单。运行期间，通过使用此右键单击功能，操作员可快速确认屏幕中所有的报警，或仅确认可见的报警、所选报警组、所选标记名，以及所选优先级。



- **抑制所选项** - 运行时操作员右击显示并指向**抑制所选项**时，可快速抑制所选的报警。
- **抑制其它项** - 操作员右击显示并指向**抑制其它项**时，可打开一个包含抑制命令的子菜单。在运行期间，通过使用此右键单击功能，操作员可快速抑制屏幕中所有的报警，或仅抑制可见的报警、所选报警组、所选标记名，以及所选优先级。



- **查询收藏夹** - 右击显示并指向**查询收藏夹**时，会出现一个**报警查询**对话框。

  如需有关详细信息，请参阅 “选择与配置报警查询收藏夹”。
- **统计** - 打开 “报警统计” 对话框。
- **抑制** — 打开 “报警抑制” 对话框。
- **冻结** - 冻结当前显示。

选择与配置报警查询收藏夹

使用 “分布式报警显示” 对象快捷菜单上的**查询收藏夹**命令，从预先定义的报警查询列表中，您可以快速选择想要显示的报警查询。此外，通过使用**查询收藏夹**命令，您可以创建新的指定查询、编辑现有查询或删除现有查询。

备注　对于 “分布式报警显示” 中出现的多行报警查询，行分隔会显示为 “乱码” 字符。但这不会影响到功能。

要选择报警查询以便显示

1. 在**运行时**使用右键单击 “分布式报警显示” 对象。

2. 单击**查询收藏夹**。此时会出现**报警查询**对话框。



3. 从当前定义的查询列表中，选择要显示的指定查询。
4. 单击**确定**。现在，“分布式报警显示”对象将显示所选指定查询的报警信息。

   如需有关动态控制 “分布式报警显示”对象的详细信息，请参阅 “分布式报警属性与函数”

另请参阅 “报警查询 QuickScript 函数”。

要添加新的指定查询

1. 在**运行时**右击 “分布式报警显示”对象。

2. 单击**查询收藏夹**。此时会出现**报警查询**对话框。

报警查询

名称	查询	从优先级	到优先级	报警状态	查询类型
缺省查询	\intouch!...	1	999	All	Summary

添加(A)... 修改(M)... 删除(D) 确定(O) 取消(C)

3. 单击**添加**。此时会出现**添加查询**对话框。



4. 在**名称**输入框中，输入要给查询使用的名称。
5. 在**查询**输入框中，输入要执行的 InTouch 报警查询集合。您可以指定一个或多个 “报警供应器”与报警组，方法与 “报警打印实用程序”的报警查询相同。

   如需有关详细信息，请参阅 “Alarm Viewer ActiveX 显示属性”。
6. 在**从优先级**字段中，输入最小优先级报警值 （1 到 999）。
7. 在**到优先级**字段中，输入报警的最大优先级值 （1 到 999）。

备注 InTouch 中配置的每个报警都有一个关联的优先级值。此值表示报警的重要程度，范围从 1 到 999，其中 1 表示最为重要。

如需有关报警优先级详细信息，请参阅 “报警优先级”。

8. 单击**报警状态**箭头，选择要在报警查询中使用的报警状态。
9. 选择要使用的**显示类型**选项。

   如需有关详细信息，请参阅 “报警摘要与报警历史”。

   另请参阅 “配置 Alarm Viewer ActiveX 控件”。
10. 单击**确定**以关闭**添加查询**对话框。
11. 在**报警查询**对话框上单击**确定**以确认添加。

要修改现有的指定查询

1. 右击 “分布式报警显示” 对象。
2. 单击**查询收藏夹**。此时会出现**报警查询**对话框。
3. 从当前定义的查询列表中，选择要修改的指定查询。
4. 单击**修改**。此时会出现**修改查询**对话框。
5. 进行必要的修改，然后单击**确定**以关闭**修改查询** 对话框。
6. 在**报警查询**对话框上，单击**确定**以确认修改。

备注 对于正在使用被修改的报警查询的其它 “分布式报警对象”，修改动作不会自动应用于它们。

要删除现有的指定查询

1. 右击 “分布式报警显示” 对象。
2. 单击**查询收藏夹**。此时会出现**报警查询**对话框。
3. 从当前定义的查询列表中，选择要删除的指定查询。
4. 单击**删除**。此时会出现一个消息框，询问您是否确定要删除所选的指定查询。
5. 单击**是**。
6. 在**报警查询**对话框上，单击**确定**以确认删除。

备注 对于正在使用要删除的报警查询的其它 “分布式报警对象”，删除动作不会自动应用于它们。

分布式报警显示准则

分布式报警显示要求在使用 “分布式报警显示” 之类的对象时，须遵循一些准则。这些准则如下：

每个显示都必须有一个标识符，以便关联的 QuickScript 函数知道需要修改哪个显示。这个标识符是在**报警配置**对话框中作为**显示名称**输入的，对于每个显示，它必须保持唯一。

显示不应覆盖其它的 InTouch 对象，如窗口控件或图形对象等。通过单击 WindowMaker 中的分布式报警显示，并检查该显示“手柄”，可以很方便地验证这点。这些手柄不应碰到屏幕上其它的图形对象。

显示不应随意使用。在屏幕上放置大量的显示，会导致系统性能下降。如有可能，应限制屏幕上的显示数量，必要时请调用含其它显示的其他屏幕（对话框）。

创建分布式报警显示

要创建分布式报警显示

1. 单击**向导 /ActiveX** 工具栏中的**向导**工具。此时会出现**向导选择**对话框。



2. 从向导列表中选择**报警显示**。
3. 双击**分布式报警显示**向导，或选择它然后单击**确定**。此时该对话框会关闭，您的窗口会再次出现，且光标处于“粘贴”模式。

4. 在窗口中单击以粘贴“报警显示”向导。

日期	时间	状态	类	类型	优先级	名称
06 十一月	11:37	UNACK	Value	HIHI	1	Alarm1
06 十一月	11:37	UNACK	Value	HI	250	Alarm2
06 十一月	11:37	UNACK	Value	LO	500	Alarm3
06 十一月	11:37	UNACK	Value	LOLO	750	Alarm4
06 十一月	11:37	ACK	Dev	Minor	1	Alarm5
06 十一月	11:37	ACK	Dev	Major	250	Alarm6
06 十一月	11:37	ACK	ROC	1	500	Alarm7

提示 要调整向导大小，请指向它的一个选择手柄，然后拖拽它直至达到所需大小。

5. 现在，您便可以开始配置显示。

配置分布式报警显示

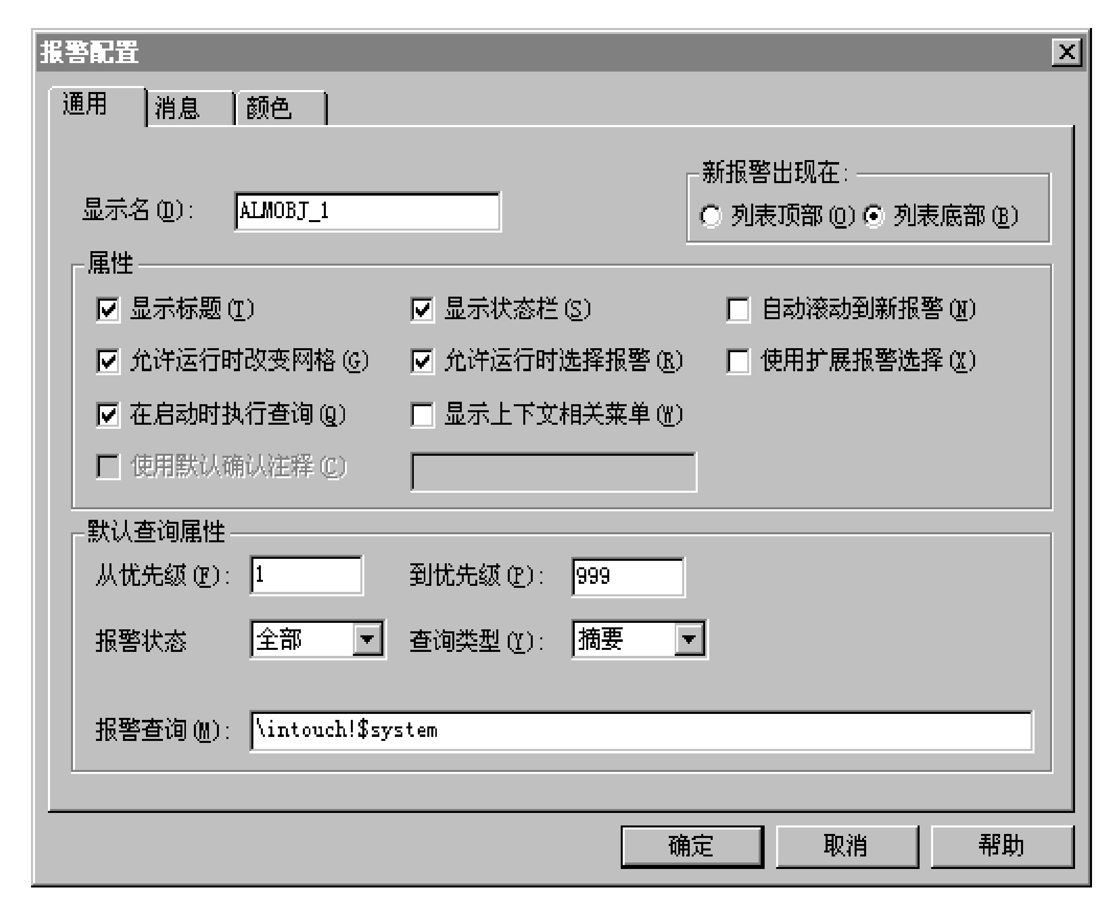
报警配置对话框有三个属性页，包括**通用**、**消息**及**颜色**等选项。

备注 配置对话框与所有的标准 Windows 属性页均很相似，在单击**确定**之前不会记录任何设置。不过，您从一个属性页（选项卡）切换到另一个属性页时，系统将核实各项内容是否正确，如果某项未通过验证，包含错误项的属性页将回到屏幕最上端，并且会出现一个指出错误的消息框。如果单击**取消**，则会忽略所有输入并关闭对话框。

分布式报警显示通用属性

要配置分布式报警显示

1. 双击分布式报警显示，或使用右键单击它，然后单击**属性**。此时会出现**报警配置**对话框，并显示**通用**属性页：



提示 如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 在**显示名**框中，输入报警显示的名称。对于所用的每个报警显示，此名称均必须保持唯一。

提示 此处输入的名称将在整个系统中用以引用此对象，以执行确认报警与查询之类的任务。

3. 选择**新报警出现在**选项，以设置新报警在对象中的显示位置：

选项	描述
列表顶部	在列表顶部显示最新的报警。
列表底部	在列表底部显示最新的报警。

4. 选择**属性**，具体描述见下文：

属性	描述
显示标题	显示报警消息标题栏。
使用默认确认注释	控制在操作员确认报警时是否使用缺省注释。如果选定了此复选框，并输入了一个字符串，则该字符串将被用作运行期间的缺省注释。如果未选定此复选框，则在操作员确认报警时，会显示一个对话框，提醒操作员输入一则注释。您既可以填写该对话框，也可以将其留为空白。
显示状态栏	显示状态栏。
允许运行时改变网格	允许用户在运行时改变列设置。
在启动时执行查询	如果选定此复选框，则会使用缺省查询属性自动开始更新显示。如果未选定，您需要在显示更新以前，执行 **almDefQuery** 或 **almQuery**。
自动滚动到新报警	如果用户从开始处滚动列表，则这会自动跳到新报警处。（这里的新报警是指那些显示对象中当前未显示的报警）。
允许运行时选择报警	允许用户在运行期间选择报警。
使用扩展报警选择	允许通过按住 Ctrl 或 Shift 键并配合使用鼠标选择多项报警。缺省情况是通过单击它们来切换选择报警（只有选定**允许运行时选择报警**复选框时，才可看到此复选框）。
显示上下文相关菜单	启用激活右击弹出式菜单的功能。

5. 选择**缺省查询属性**选项，具体描述见下文：

提示 如果您选择**在启动时执行查询**选项，或执行 **almDefQuery** QuickScript 函数，则会使用**默认查询属性**。

属性	描述
从优先级	缺省的最小报警优先级。
到优先级	缺省的最大报警优先级。如需有关报警优先级的详细信息，请参阅本章的“报警优先级”部分。
报警状态	缺省报警状态查询（“全部”、“撤消确认”、“确认”）。

属性	描述
查询类型	将显示类型设置为 “摘要”或 “历史”。
报警查询	设置初始的报警查询。此字段只接受文本；它不接受标记。这些列表的有效语法包括： **\\Node\InTouch!Group** “报警组”的完整路径 **\InTouch!Group** 本地 “报警组”的完整路径 **GroupList** 另一个 “组列表”

备注 若要执行多项查询，请用空格将每项查询分开。

例如：**\\Master\InTouch!MyGroup LocalGroupList**

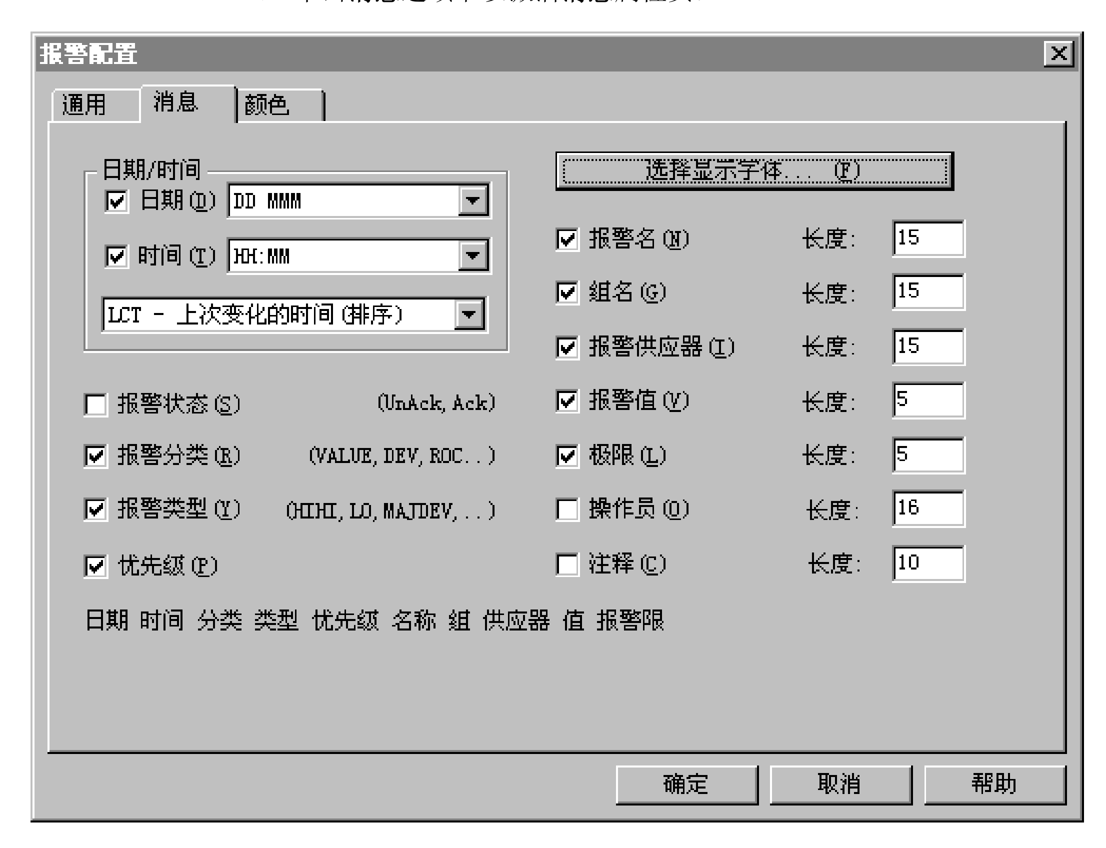
分布式报警消息格式

分布式报警显示对象中显示的信息是可以配置的。例如，要显示的信息，以及在某些情况下要给某个项目显示的字符数等。

要配置报警显示消息的格式

1. 双击所需的 DAO。此时会出现**报警配置**对话框。

2. 单击**消息**选项卡以激活**消息**属性页：



提示 如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

预览区 （对话框底部）会显示当前配置的报警消息的一个实例。此示例使用所选字体来显示消息，但不包括颜色。

3. 单击**日期格式**日期格式以选择日期格式。可用的格式包括：

选择	显示	选择	显示
DD MMM	28 Feb	**MM/DD**	02/28
DD MM YYYY	28 Feb 2002	**MM/DD/YY**	02/28/02
DD/MM	28/02	**MMM DD**	Feb 28
DD/MM/YY	28/02/02	**MMM DD YYYY**	Feb 28 2002
YY/MM/DD	02/02/28	**YYYY/MM/DD**	2002/02/28

4. 单击**时间格式**箭头以选择时间格式。此字段中的值将用作指定时间格式的模板。例如，若要将时间指定为 **10:24:30 AM**，请使用 **HH:MM:SS AP**。模板字符如下：

AP	选择 AM/PM 格式。例如，下午三点钟显示为 3:00 PM。非此格式的时间缺省使用 24 小时军用时间格式。例如，下午三点钟显示为 15:00。
HH	显示报警 / 事件发生时刻的小时数。
MM	显示报警 / 事件发生时刻的分钟数。

SS	显示报警 / 事件发生时刻的秒数。
SSS	显示报警 / 事件发生时刻的毫秒数。

5. 使用**显示的时间**下拉菜单下的单选钮，选择您希望报警在报警对象中的排序方式。总共有两个选项：

OAT	原始报警时间 - 即报警发生的日期 / 时间标签。
LCT	上次更改时间 - 即报警实例最近一次改变状态的日期 / 时间标签：包括报警发生、子状态改变、恢复正常或确认。

6. 单击**确定**。

列管理按钮

“列明细”对话框可通过单击**通用**属性页上的**列管理**按钮来调用。“列详细”对话框用于选择要显示的列、指定显示顺序，以及设置列名和宽度。



要配置显示列明细

1. 双击分布式报警显示，或使用右键单击它，然后单击**属性**。此时会出现**报警配置**对话框，并显示**通用**属性页。
2. 单击**消息**选项卡。
3. 单击**列管理**按钮。此时会出现 “列明细”对话框。
4. 从**列明细**对话框中，选择**列名**旁的复选框，以便在报警对象中显示该列。可供您选择显示的列名现介绍如下。

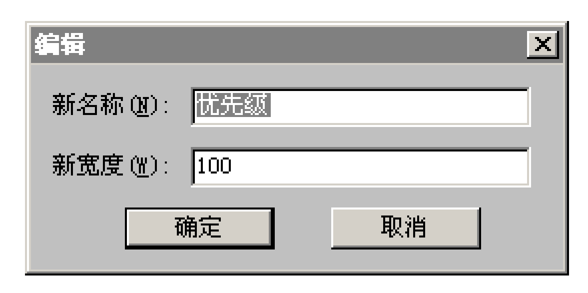
备注 您至少应选择一个列。

列	描述
日期	按**消息**属性页中选定的格式显示日期。
状态	显示报警的状态。

列	描述
时间	按**消息**属性页中选定的格式显示时间。
名称	显示报警 / 标记名。
描述	显示报警的描述。
组	显示 “报警组”名称。
类型	显示报警类型。
值	显示报警发生时标记名的值。列宽应该足够大以满足精度要求。
报警限	显示标记名的报警限值。列宽应该足够大以满足精度要求。
优先级	显示报警优先级。
类	显示报警类别。
供应器	显示报警供应器的名称。
操作员	显示与报警条件关联的已登录操作员的 ID。
UTC 时间	显示报警的 UTC 时间，也称为 “格林尼治标准时间”、“通用协调时间”或祖鲁时间。
注释	显示标记名的注释。这些注释是在数据库中定义标记名时输入 “报警注释”框的。

备注 缺省情况下，除**操作员**与**注释**外，所有列名均被选择。

5. 要重新整理列，请选择列名，然后使用 “上移”与 “下移”箭头键。**列明细**对话框顶部出现的列名即为在报警显示最左边显示的列。
6. 要编辑列名与宽度，请选择列名然后单击**编辑**。此时会出现该列的**编辑**对话框。



7. 如果您要显示与缺省列名不同的列名，请在**新名称**文本框中输入新的名称。
8. 在**新宽度**文本框中输入列宽。列宽按像素计算，范围是从 1 至 999 像素。缺省列宽为 100 像素。
9. 单击**编辑**对话框上的**确定**。
10. 单击**列明细**对话框上的**确定**。

11. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

分布式报警显示颜色属性

要配置报警显示颜色

1. 双击分布式报警显示对象，或使用右键单击它，然后单击**属性**。此时会出现**报警配置**对话框。
2. 单击**颜色**选项卡以激活**颜色**属性页：



提示 如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

3. 在顶部的**显示**组中，单击每个颜色方框以打开调色板。在调色板上为下列各项单击您要使用的颜色：

选项	描述
窗口	设置显示背景颜色。
网格	设置显示网格颜色。
选择背景	设置突出显示文本背景颜色。
选择文本	设置突出显示文本颜色。
标题栏背景	设置标题栏背景颜色 （仅当 “显示标题”选项为开时此项才可见）。

选项	描述
标题栏文本	设置标题栏文本颜色（仅当“显示标题”选项为开时此项才可见）。
报警返回	设置返回的报警颜色（未经确认返回到正常状态的报警）。
事件	设置事件报警颜色。

4. 在**报警优先级**框中，输入用于报警显示的断点值。
5. 单击**撤消确认报警**与**确认报警**颜色框以打开 InTouch 调色板。单击调色板中您要使用的颜色。
6. 单击**确定**。

配置显示报警查询

分布式报警显示可以显示活动报警的摘要或报警历史的列表。您可以动态地改变是显示报警摘要，还是显示报警历史。

要更改分布式报警对象查询缺省值

1. 双击分布式报警对象，或使用右键单击它，然后单击**属性**。此时会出现**报警配置**对话框，并显示**通用**属性页：



提示 如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 单击**查询类型**箭头，并选择您要将其用作运行时缺省值的报警显示类型。

   例如，在运行期间，显示类型是由您使用的查询 QuickScript 函数确定的。如果针对 *AlarmObj_2* 显示对象运行 *almQuery( )* QuickScript，且 *Type* 参数设置为 “摘要”，则显示对象将显示当前报警的摘要信息。相反，针对同样的显示对象，如果在运行 *almQuery( )* 时将 *Type* 参数设置为 “历史”，则它将显示报警历史。*QueryType* 属性反映了报警显示的当前状态。

3. 单击**确定**。

使用分布式显示监视本地报警

分布式报警显示可用于显示与确认本地及远程报警。

要设置仅监视本地报警的显示

1. 将分布式报警显示对象粘贴到窗口。（单击向导工具，选择**报警显示**类别。双击**分布式报警显示**向导，然后单击窗口以便粘贴它）。
2. 双击显示对象，或右击它然后单击**属性**。此时会出现**报警配置**对话框，并显示**通用**属性页。
3. 在**报警查询**框中，输入 **\InTouch!$System**。

> **提示** 您可以将 **$System** 替换成任何有效的 “报警组”。您还可以定义一个仅包含 **\InTouch!$System** 的 “报警组列表”，然后在步骤 3 中使用此 “组列表”而不是直接引用。

4. 针对显示类型与应用程序要求的过滤设置，配置**默认查询属性**的其它参数。
5. 切换到 WindowViewer 以运行应用程序。

分布式报警属性与函数

“分布式报警显示”包含多个标记名**点域**和 QuickScript 函数。下面将简单介绍如何使用 QuickScript 函数。

有关如何使用 QuickScript 函数的详细信息和示例，请参阅联机 *InTouch 参考指南*。

报警显示监视属性

报警显示有多个采用 QuickScript 的属性，可用于在运行时监视显示的状态。这些属性可通过 *GetPropertyX( )* 函数进行访问，其中， X 表示数据类型 （D 表示离散型， I 表示整型， M 表示消息型）。

查询属性	描述
.AlarmGroup	包含当前查询列表的消息型属性。
.PriFrom	包含当前查询优先级低过滤值的整型属性。

查询属性	描述
.PriTo	包含当前查询优先级高过滤值的整型属性。
.QueryType	包含当前查询类型的整型属性： **1 = 历史 2 = 摘要**
.QueryState	包含当前查询过滤器的整型属性： **0 = 全部 1 = 未确认 2 = 确认**

如需有关详细信息，请参阅 “配置 Alarm Viewer ActiveX 控件”。

查询状态优先级	描述
.Successful	包含当前查询状态的离散属性： **0 = 错 1 = 对**
.ProvidersReq	包含当前查询中的报警供应器数的整型属性。
.ProvidersRet	包含成功返回查询结果的报警供应器数的整型属性。
.NumAlarms	包含当前查询中的报警数的整型属性。
.PageNum	包含报警显示中显示的当前页码的整型属性。
.TotalPages	包含报警显示中的总页数的整型属性。

报警显示控制属性

报警显示也有两个采用 QuickScript 的属性，可用于控制**运行时**显示屏幕的移动。这些属性通过 *SetPropertyD* 函数来控制。

控制属性	描述
.NextPage	此属性从 0 转变为 1 时，报警显示向下滚动一页。
.PrevPage	此属性从 0 转变为 1 时，报警显示向上滚动一页。

只要此离散变量的值从 “关”（0 或 False）变为 “开”（1 或 Ture）时，报警显示对象都会显示与该 QuickScript 对应的页 （下一页或上一页）。显示该页之后，该离散型变量将自动复位为 “关”（0 或 False）。

备注 提供这些函数是为了简化从以往的 InTouch 标准显示转换到当前的分布式显示的工作。它们的功能已经被滚动条和 *almMoveWindow* QuickScript 函数所取代。

报警确认 QuickScript 函数

“分布式报警系统”能够确认它可以查询到的任何报警（仅限于摘要显示）。为提供此项功能，“分布式报警系统”加入了报警确认 QuickScript 函数。这些函数是对 InTouch 报警系统用于确认本地报警和“报警组”的 **.Ack** 点域的补充。如需有关这些函数具体语法的详细信息，请参阅 *InTouch 参考指南*。

函数	描述
almAckAll	确认当前报警查询中的所有报警。由于报警显示的显示区域有限，所以 **almAckAll** 函数可能会确认显示中不可见的那些报警。
almAckDisplay	仅确认报警显示中当前可见的那些报警。
almAckGroup	确认来自相同报警供应器且具有给定组名的所有报警。
almAckPriority	确认具有相同报警供应器名与组名且优先级在指定范围内的所有报警。
almAckRecent	仅确认当前报警查询中最近发生的报警。
almAckSelect	分布式报警显示允许在运行时单击鼠标来选择报警。**almAckSelected** 函数可用于确认这些报警。
almAckSelectedGroup	确认与一个或多个所选的报警具有相同组名且来自相同供应器的所有报警。
almAckSelectedPriority	确认来自相同供应器与组名且与一个或多个所选的报警具有相同优先级的所有报警。
almAckSelectedTag	确认来自相同供应器与组名、使用相同的标记名且与一个或多个所选的报警具有相同优先级的所有报警。
almAckTag	确认在给定优先级范围内具有相同 “标记名”、“供应器”名及 “组”名的所有报警。

报警抑制 QuickScript 函数

“分布式报警系统”能够通过确立一套排除标准，在“报警接收器”端抑制一个或多个报警。如果报警与排除标准匹配，则它在“报警接收器”端会是不可见的，它不会出现在显示中、不能打印，也不能在特定的“报警接收器”端记录。为提供此项功能，“分布式报警系统”加入了报警抑制 QuickScript 函数。如需有关这些函数具体语法的详细信息，请参阅 *InTouch 参考指南*。

函数	描述
almSuppressAll	抑制显示所有当前活动报警的当前与将来的所有实例。
almSuppressSelected	抑制显示所选报警的当前与将来的实例。
almSuppressDisplay	抑制显示任何可见报警的当前与将来的实例。
almSuppressSelectedGroup	抑制显示与一个或多个具有相同“供应器”名的所选报警属于相同报警组的任何报警的当前与将来的实例。
almSuppressSelectedPriority	抑制显示与一个或多个具有相同“供应器”名与“组”标记的所选报警具有相同优先级的当前与将来的显示。
almSuppressSelectedTag	抑制显示特定的所有报警的当前及未来实例，这些报警与一个或多个选定的报警具有相同的“标记名”名称，这些选定的报警具有相同的“供应器名”、“组名”及“优先级”范围。
almSuppressGroup	抑制显示属于给定组名且使用相同“供应器”名的任何报警当前与将来的实例。
almSuppressPriority	抑制显示指定优先级范围内所有报警的当前及未来的实例，这些报警具有相同的“供应器名”和“组名”。
almSuppressTag	抑制显示通过给定标记名来确定的所有报警的当前及未来的实例，这些报警具有相同的“供应器名”、“组名”及“优先级”范围。
almUnSuppressAll	清除报警抑制。
almSuppressRetain	更改报警查询时在报警查询之间保持报警抑制。

报警显示控制 QuickScript 函数

“分布式报警系统”提供了若干个 QuickScript 函数来处理显示对象。这些函数可供移动显示窗口、在显示内选择报警、显示选定报警的数目，以及显示统计窗口。

如需有关这些函数具体语法的选项信息，请参阅联机 *InTouch 参考指南*。

函数	描述
almMoveWindow	提供用于控制显示窗口的命令。这些命令包括：向上翻页 (Page Up)、向下翻页 (Page Down)、向右滚动 (Scroll Right)、向左滚动 (Scroll Left)、向上一行 (Line Up)、向下一行 (Line Down)、顶部 (Top)、底部 (End) 等。
almSelectAll	切换选择显示中的所有报警。由于报警显示的显示区域有限，所以 **almSelectAll** 函数可能会选择显示中不可见的那些报警。
almSelectionCount	返回包含操作员在 “分布式报警对象”中所选报警数的整型值。
almSelectGroup	切换选择给定 “供应器”中来自给定 “报警组”的所有报警。
almSelectItem	切换在报警显示中辉亮显示的选定项目。
almSelectPriority	切换选择给定 “供应器”、“报警组”内优先级在给定范围内的所有报警。
almShowStats	显示报警统计对话框。
almSelectTag	切换选择特定报警 “供应器 / 组 / 标记”内的所有报警。您也可以指定 “优先级”范围，或使用 1-999。
almUnSelectAll	取消选择所有选定的记录。

报警查询 QuickScript 函数

分布式显示通过提交报警查询来检索报警信息。此查询的参数与查询类型在三个 QuickScript 函数之一中指定。如需有关这些函数具体语法的详细信息，请参阅 *InTouch 参考指南*。

函数	描述
almDefQuery	使用 “配置”对话框的缺省属性执行查询。这些属性包括：“从优先级”、“到优先级”、“报警列表”以及 “显示类型”。缺省属性只能在开发时更改，并且不会被其它报警查询改写。
almQuery	执行摘要或历史报警信息查询。此函数中提供了所有的查询属性。
almSetQueryByName	使用与用户定义的查询名关联的查询参数开始一项新的报警查询。

备注 在 Windows 2000 中，如果 “报警供应器” 与 “报警接收器” 在不同的域中，则 “报警接收器” 将无法查看报警，除非查询使用供应器设备的全限定名称或供应器名称的 IP 地址。例如，可以在查询中将另一个域中的 “报警供应器” 指定为：
\\provider1.b3.wonderware.com\intouch!$system
其中， "provider1" 是机器名， "b3.wonderware.com" 是域的 “主 DNS 后缀”。

Alarm DB View ActiveX 控件

InTouch 提供了一个 Alarm DB View ActiveX 控件，供您以可视化方式直观地查看新的 Alarm DB Logger 数据库中的报警数据。此控件可用于查看所有的报警和事件信息。通过指定下列属性，您可以设计 ActiveX 控件和所显示数据的外观。

- 上下文相关菜单功能
- 显示模式
- 列表控制选项
- 不同属性的颜色
- 字体类型、样式和大小
- 数据库规格 （服务器名、用户 ID 及口令）
- 查询过滤器
- 列管理
- 排序

设计好控件格式之后，用户将能够进行以下调整，以操纵他们所查看的数据：

- 给列中的信息排序
- 更新显示
- 执行查询
- 调整列宽

您可以将 Alarm DB View ActiveX 控件放入 WindowMaker，并调整它的大小和位置；您可以使用 “属性” 页来配置控件。随后，您便可以在 Alarm DB View ActiveX 控件的视图窗口中查看数据库中的数据。所查看数据的类型和格式取决于 “属性” 页中设置的属性。

安装

Alarm DB View ActiveX 控件随 InTouch 一起安装。

要在 WindowMaker 窗口中粘贴 Alarm DB View ActiveX 控件

1. 打开 “向导选择” 对话框。
2. 选择 ActiveX 控件下的 **AlmDbViewCtrl**，然后单击**确定**。

3. 将控件粘贴到窗口上，并调整到所需的大小。

卸载

1. 删除粘贴到窗口上的所有 Alarm DB View 控件。
2. 从**特别**菜单上，选择 “配置”。
3. 选择**向导 /ActiveX 安装**，然后打开 “向导 /ActiveX 安装”对话框。
4. 选择 **ActiveX 控件**安装属性页。 Wonderware Alarm Database View 控件名将显示于“安装的 ActiveX 控件”文本区。
5. 单击 Wonderware Alarm Database View 控件，然后单击**删除**。
6. 在显示警告消息时单击**是**。
7. 单击**关闭**。

访问 **AlmDbViewCtrlX** 属性对话框

您可以通过下列方法来访问 “AlmDbViewCtrlX 属性”对话框：

- 双击控件，或
- 右击控件，然后从弹出式菜单中选择 “属性”菜单。

AlmDBViewCtrlX 属性对话框

“AlmDbViewCtrlX 属性”对话框上的选项卡提供了许多选项，可用于设计控件的属性。单击下面适当的选项卡 ID 可查看到有关填写选项卡上各个字段的操作说明。

备注 下文所提供的操作说明通过将 “AlmDbViewCtrlX 属性”对话框上的属性通可以在脚本编辑器中配置的那些属性关联起来，来介绍特定控件的各种属性。

例如：
#AlmDBView1.RefreshMenu = False;

此表达式指设置 AlmDBView1 对象的属性值。

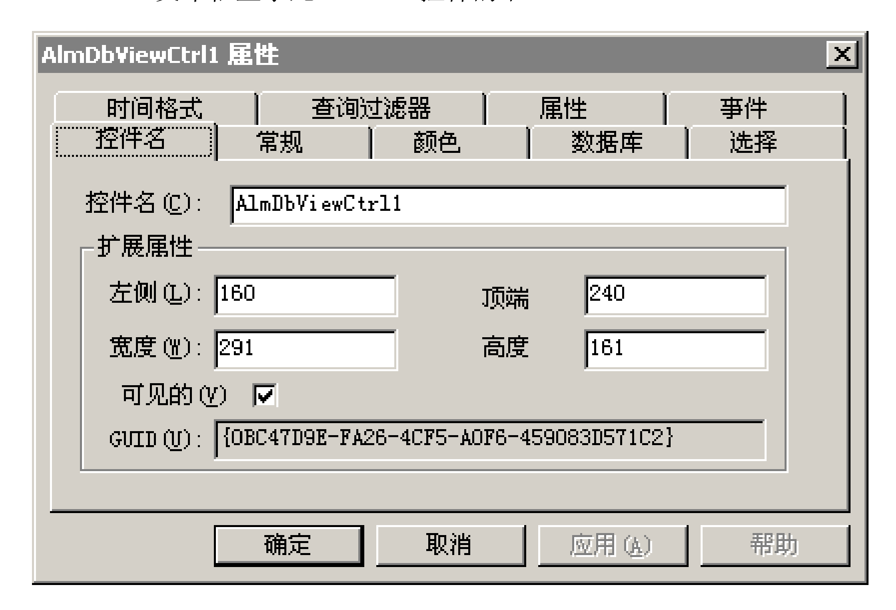
在下面数节，语法会使用 Object.RefreshMenu 之类的表示法进行描述。在所有的例子中， "Object" 指相当于对象或控件的对象表达式。

控件名页

- “属性”对话框的**控件名**选项卡提供控件与应用程序有关的详细信息。

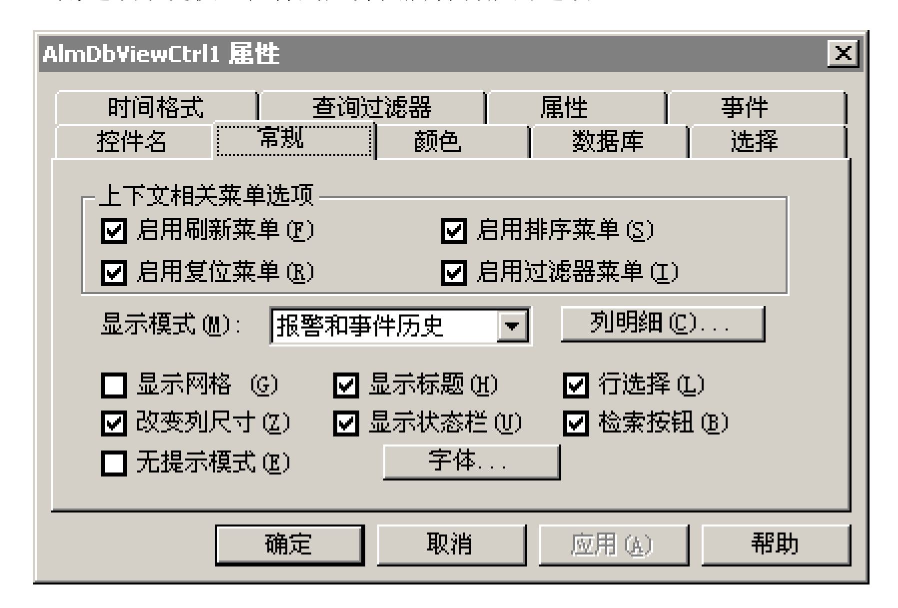
备注 在缺省条件下，“控件名”由该控件的 ProgID 确定。 ProgID 是在计算机中安装 ActiveX 控件时输入系统注册表的名称。将该控件的实例放入 InTouch 应用程序时，程序会从系统注册表读取该控件的 ProgID，并在其后附加一个索引号，结果就生成了控件名，如 **AlarmDBViewCtrl1**。

- **扩展属性**部分介绍窗口中控件的配置 （左侧、顶部、宽度及高度）信息；您可以更改这些属性。
- 取消选择**可见的**复选框可使得控件在运行期间不可见。
- **GUID** 文本框显示此 ActiveX 控件的唯一 ID。



通用属性页

通用选项卡提供此控件用户界面所含功能的选项。



上下文相关菜单选项

“上下文相关菜单选项”区域提供的选项可启用和禁用此控件中可用的上下文相关菜单功能。

下面关于 AlmDbViewCtrl 属性和方法的描述基于下述假设 （用在编写 InTouch 脚本时）：

给含 “布尔”值的整型标记指定值 "True"，并且读取整型标记名，或是从整型标记名获取时：

设置：指定一个不等于 0 或 "True" 的值
获取：您会获取整数值 -1

给含 “布尔”值的整型标记指定值 "False"，并且读取整型标记名，或是从整型标记名获取时：

设置：指定 0 或 "False"
获取：您会获取整数值 0

设置示例：

```
#AlmDbViewCtrl6.SortMenu = "True"; { 在控件的上下文菜单中启用
“排序”菜单 }

#AlmDbViewCtrl6.SortMenu = 1; { 在控件的上下文菜单中启用 “排序”
菜单 }

#AlmDbViewCtrl6.SortMenu = "False"; { 在控件的上下文菜单中禁用
“排序”菜单 }

#AlmDbViewCtrl6.SortMenu = 0; { 在控件的上下文菜单中禁用 “排序”
菜单 }
```

获取示例：

```
IF #AlmDbViewCtrl6.ShowFetch == 0 THEN { 检查属性的实际状态 }

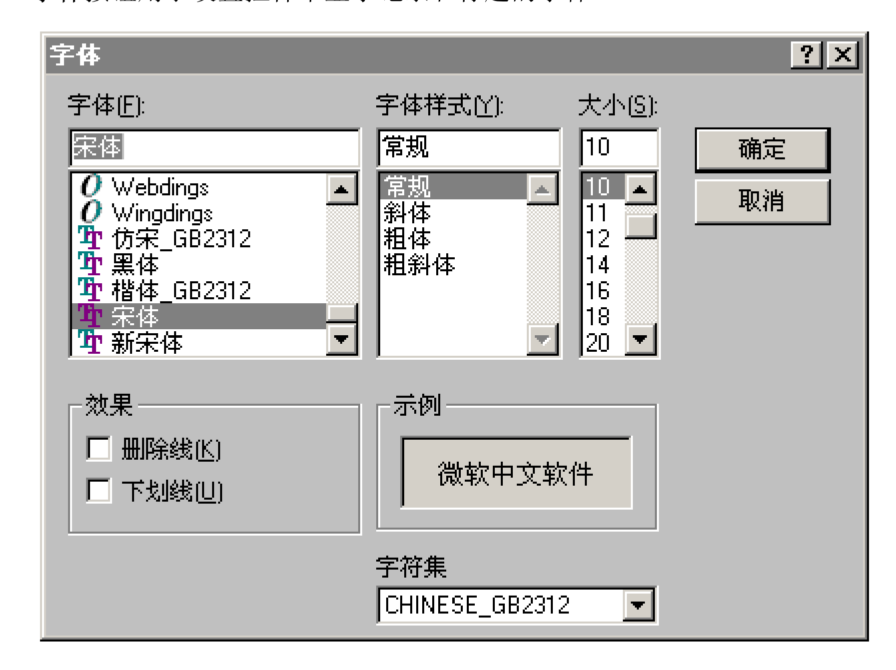
#AlmDbViewCtrl6.ShowFetch = 1; { 如果属性先前的状态是 "False"，
则将它的状态切换为 "True"}

ELSE
#AlmDbViewCtrl6.ShowFetch = 0; { 如果属性先前的状态是 "True"，
则将它的状态切换为 "False"}

ENDIF;
Int_Tag = #AlmDbViewCtrl6.ShowGrid { 显示网格时将 -1 指定给
Int_Tag   否则，将 0 指定给它 }
```

字体按钮

字体按钮用于设置控件中显示记录和标题的字体。



要配置字体属性

1. 双击 AlarmDBView ActiveX 控件，或右击它，然后单击**属性**。此时会出现 **AlarmViewerCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**通用**选项卡，然后单击**字体**按钮。此时会显示 “字体”对话框。
3. 滚动**字体**列表，然后选择所需的字体类型。
4. 滚动**字体样式**列表，以选择一种字体样式。
5. 滚动**大小**列表，以选择一种字体大小。
6. 在**效果**区，选中**删除线**复选框或**下划线**复选框，以选择 “删除线”或 “下划线”属性。
7. 单击**字符集**下列箭头以选择所需的脚本类型。

备注 **样本**框会显示所选字体属性的示例。

8. 单击**确定**。

启用刷新菜单

选择此复选框以启用控件右击菜单的**刷新**菜单选项。“刷新”菜单会将控件刷新到数据库，如果连接成功，它会显示范围 1 到 MaxRecords 之间的记录集。

属性：	**RefreshMenu**		
用途：	返回或设置一个值，该值确定上下文相关菜单中是否显示“刷新”菜单项。		
类型：	离散		
缺省值：	True		
语法：	*Object*.RefreshMenu *[= Integer]*		
	离散		
		True	（缺省值）显示“刷新”菜单项
		False	不显示“刷新”菜单项。

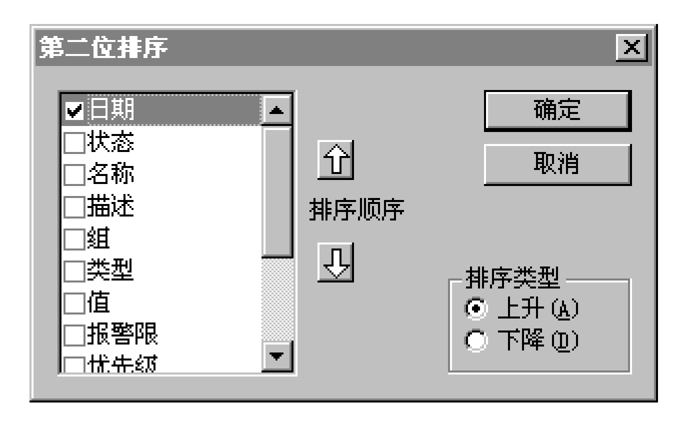
启用排序菜单

选择此复选框以启用控件右击菜单的**排序菜单**选项。此菜单显示“辅助排序”菜单，用于设置用户自定义的列排序方式。

属性：	**SortMenu**		
用途：	返回或设置一个值，该值确定上下文相关菜单中是否显示“排序”菜单项。		
类型：	整型		
缺省值：	True		
语法：	*Object*.SortMenu *[= Integer]*		
	整型	整型表达式，指定是否按“设置”所述显示“排序”菜单项。	
		True	（缺省值）显示“排序”菜单项
		False	不显示“排序”菜单项。

辅助排序

单击此弹出式菜单时，会显示**辅助排序**对话框。



此对话框显示控件中当前显示的列清单。此对话框可用于按升序或降序对单个列和多个列进行排序。

要指定给哪些列排序，请选择列名旁边的复选框。使用**排序顺序**箭头键可以重新整理各个列。

例如，如果要先按报警状态，然后按日期的降序进行排序：

1. 选择这两个复选框 （"日期"和 "状态"）。
2. 选择**状态**行。
3. 单击 "向上"**排序顺序**箭头键。
4. 从**排序类型**区选择**降序**单选钮。
5. 单击**确定**。

启用过滤器菜单

选择此复选框以启用控件右击菜单的**过滤器**菜单选项。此菜单显示过滤器菜单，用于设置用户定义的过滤标准。

属性：	**FilterMenu**		
用途：	返回或设置一个值，该值确定上下文相关菜单中是否显示 "过滤器"菜单项。		
类型：	离散		
缺省值：	`True`		
语法：	*Object*.FilterMenu *[= Discrete]*		
		True	（缺省值）显示 "过滤器"菜单项
		False	不显示 "过滤器"菜单项。

启用复位菜单

此复选框用于启用控件右击菜单的**复位菜单**选项。“复位”菜单可以按照设计时保存的设置排列所有的列。

属性：	**ResetMenu**		
用途：	返回或设置一个值，该值确定上下文相关菜单中是否显示“复位”菜单项。		
类型：	离散		
缺省值：	True		
语法：	*Object*.ResetMenu *[= Discrete]*		
		True	（缺省值）显示“复位”菜单项
		False	不显示“复位”菜单项。

显示模式

此下拉列表框显示一个可用视图选项的列表。列表的内容包括：

- 报警与事件历史
- 报警历史
- 事件历史。

属性：	**DisplayMode**
用途：	返回控件的显示模式。此属性为**只读**属性。
类型：	字符串
缺省值：	报警与事件历史
语法：	*Object*.DisplayMode
	`tagname = #AlmDbView1.DisplayMode;`
	其中，控件的名称是 AlmDbView1，标记名定义为“消息”型标记名。

运行时功能复选框

显示网格复选框

显示网格复选框用于启用或禁用显示控件网格的功能。

属性：	**ShowGrid**		
用途：	返回或设置一个值，该值确定是否在控件中显示网格线。		
类型：	离散		
缺省值：	`False`		
语法：	*Object*.ShowGrid *[= Discrete]*		
		True	在控件中显示网格线。
		False	（缺省值）不显示网格线。

显示标题复选框

显示标题复选框用于启用或禁用显示控件标题的功能。

属性：	**ShowHeading**		
用途：	返回或设置一个值，该值确定是否在控件中显示列标题。		
类型：	离散		
缺省值：	`True`		
语法：	*Object*.ShowHeading *[= Discrete]*		
		True	（缺省值）在控件中显示列标题。
		False	不显示列标题。

- 主排序

  标题给用户提供了主排序选项。单击标题时，会对满足上次查询条件的所有行进行排序，并且给用户显示排序结果中的第一条记录。

- 移动列

  通过单击并拖动列标题，可以在控件中移动列，并可以将它放到靠近其它某些列的位置。

无提示模式

选择此复选框可启用 “无提示模式”选项。此属性确定控件是否处于无提示模式。选定此复选框时，不会显示任何错误消息。要查看错误，请调用 "GetLastError" 方法来返回错误消息。

属性：	**SilentMode**		
用途：	返回或设置一个值，该值确定控件是否处于无提示模式。		
类型：	离散		
缺省值：	`False`		
语法：	*Object*.SilentMode *[= Discrete]*		
		True	处于无提示模式。
		False	（缺省值）关闭无提示模式。

行选择复选框

行选择复选框用于启用或禁用控件的 “行选择”选项。

属性：	**RowSelection**		
用途：	返回或设置一个值，该值确定是否允许在运行时选择行。		
类型：	离散		
缺省值：	`True`		
语法：	*Object*.RowSelection *[= Discrete]*		
		True	（缺省值）允许在控件中选择行。
		False	不允许选择行。

备注 如果不允许选择行 - 即该属性被设置为 `False` - 则不会生成 “单击”或 “双击”事件。

更改列尺寸复选框

更改列尺寸复选框用于启用或禁用控件的列尺寸调整选项。

属性：	**ColumnResize**
用途：	返回或设置一个值，该值确定运行时是否可以调整列的大小。
类型：	离散

属性：	**ColumnResize**		
缺省值：	`True`		
语法：	*Object*.ColumnResize *[= Discrete]*		
		True	（缺省值）运行时可以调整列大小。
		False	不能调整列大小。

显示状态栏复选框

显示状态栏复选框用于启用或禁用在控件底部显示状态栏的功能。

此图形用户界面功能同以下属性有关：

属性：	**ShowStatusBar**		
用途：	返回或设置一个值，该值确定是否显示状态栏。		
类型：	离散		
缺省值：	`True`		
语法：	*Object*.ShowStatusBar *[= Discrete]*		
		True	（缺省值）显示状态栏。
		False	不显示状态栏。

状态栏

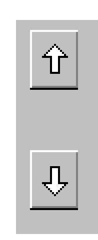
状态栏显示控件的当前状态。

(local)	显示 1 到 34 (在 34 记录中)	已连接的

左侧窗格显示服务器名 （属性是 ServerName），右侧窗格显示服务器的连接状态 （属性是 ConnectStatus）。中间窗格显示符合查询条件的总记录数 （属性是 TotalRowcount）中当前显示的记录数 （属性是 RowCount）。

检索按钮复选框

检索按钮复选框用于启用或禁用在控件右侧显示检索按钮的功能。



属性：	**ShowFetch**		
用途：	返回或设置一个值，该值确定是否显示检索按钮。		
类型：	离散		
缺省值：	`True`		
语法：	*Object*.ShowFetch *[= Discrete]*		
		True	（缺省值）显示检索按钮。
		False	不显示检索按钮。

列明细按钮

“列明细”对话框可通过单击**通用**属性页上的**列明细**按钮来调用。“列详细”对话框用于选择要显示的列、指定显示顺序，以及设置列名和宽度。



要配置显示列明细

1. 双击分布式报警显示，或使用右键单击它，然后单击**属性**。此时会出现 **AlarmDBViewCtrl** 对话框，并显示**控件名**属性页。
2. 单击**通用**选项卡以激活**通用**属性页。
3. 单击**列明细**按钮。此时会出现 “列明细”对话框。
4. 从**列明细**对话框中，选择**列名**旁的复选框，以便在报警对象中显示该列。**列明细**对话框中的列有**名称**、**宽度**及**原名**。**原名**显示在发生更改之前，这些列的名称是什么。下面介绍您可以选择显示原始的列名。

备注 您至少应选择一个列。

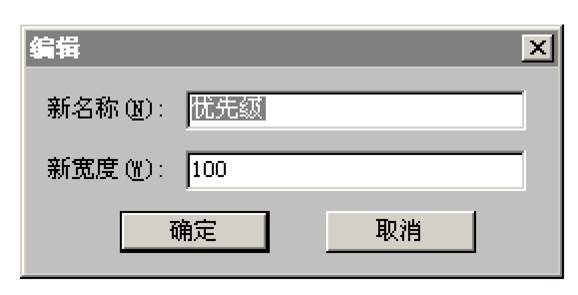
列	描述
时间	按 “时间格式”属性页中选定的格式显示时间。
状态	显示报警状态。
名称	显示报警 / 标记名。
描述	显示报警描述。
组	显示 “报警组”名称。
类型	显示报警类型。
值	显示报警发生时标记名的值。列宽应该足够大以满足精度要求。
报警限	显示标记名的报警限值。列宽应该足够大以满足精度要求。
优先级	显示报警优先级。
类别	显示报警类别。
供应器	显示报警供应器的名称。
操作员	显示与报警条件关联的已登录操作员的 ID。
域名	显示报警的域名。
用户全名	显示登录用户的全名。
持续时间	显示报警持续时间。
用户 **1**	显示与报警对应的用户定义的数字 1 的数值。
用户 **2**	显示与报警对应的用户定义的数字 2 的数值。

列	描述
用户 3	显示与报警对应的用户定义的字符串的字符串值。
注释	显示标记名的注释。这些注释是在数据库中定义标记名时输入“报警注释”框的。在给出“确认注释”时，这些注释也会更新。

备注 缺省情况下，除**操作员**与**注释**外，所有列名均被选择。

提示 如果显示模式发生更改，则列名会被重置为默认列名。在更改列名之前，最好是先选择显示模式。

5. 若要重新整理列，请选择列名，然后使用“上移”与“下移”箭头键。**列明细**对话框顶部出现的列名即为在报警显示最左边显示的列。
6. 若要编辑列名和宽度，请双击列名或选择一个列名，然后单击**编辑**。此时会出现该列的**编辑**对话框。



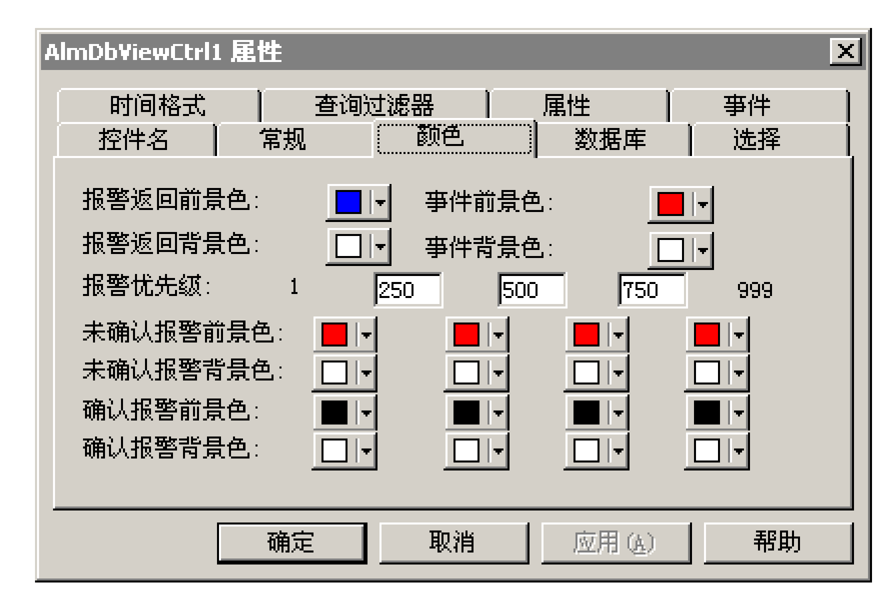
7. 如果您要显示与缺省列名不同的列名，请在**新名称**文本框中输入新的名称。
8. 在**新宽度**文本框中输入列宽。列宽按像素计算，范围是从 1 至 999 像素。缺省列宽为 100 像素。
9. 单击**编辑**对话框上的**确定**。

备注 单击**重置为默认值**以使之回到缺省的**列明细**设置。

10. 单击**列明细**对话框上的**确定**。
11. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

颜色页

颜色选项卡用于设置各种报警与事件记录的颜色。



属性

属性列表显示一组可设置颜色的记录明细。

若要设置属性，请输入 #object.PropertyName = 1; 或 #object.PropertyName = tag1; 其中， object 是 AlarmDBViewCtrl 的名称， tag1 是一个离散型标记。例如，若要设置 AckRtnForeColor1 属性，请输入 #AlarmDBView1.AckRtnForeColor1 = 1。

若要获取属性，请输入 tag1 = #object.PropertyName; 其中， object 是 AlarmViewer 的名称， tag1 是一个离散型标记。例如，若要获取 AckRtnForeColor1 属性，请输入 tag1 = #AlarmDBView1.AckRtnForeColor1;。

InTouch 可以接受英文双引号括起的单词 "True" 和 "False"，并将它们分别当作值 1 和 0。例如，动作脚本：

#AlmDbViewCtrl17.FilterMenu = "False";

会从 Alarm DB View 控件的右击菜单中删除 “过滤器菜单”。

列表包含下列属性：

属性：	**AckOrAlarmDuration**
用途：	持续时间列会填入 “确认持续时间” 或 “报警持续时间”。如果是 “确认持续时间”，则为 FALSE (0)，如果是 “报警持续时间”，则为 TRUE (1)。
类型：	Integer
缺省值：	False
语法：	*Object*.AckOrAlarmDuration *[= integer]*

属性：	**AlmRtnForeColor**	
用途：	返回或设置已返回报警的前景颜色。此颜色会应用于控件中所显示的状态为 ALM_RTN 且优先级在范围 1 到 ColorPriorityRange1 之间的记录。	
类型：	整型	
缺省值：	绿色	
语法：	*Object*.AckRtnForeColor1 *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**AlmRtnBackColor**	
用途：	返回或设置未确认报警的背景颜色。此颜色会应用于控件中所显示的状态为 ALM_RTN 且优先级在范围 1 到 ColorPriorityRange1 之间的记录。	
类型：	整型	
缺省值：	绿色	
语法：	*Object*.AckRtnForeColor1 *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性:	**ColorPriorityRange1**
用途:	设置要显示的报警的优先级范围之边界。此属性的值必须大于 1 且小于 ColorPriorityRange2。
类型:	Integer
缺省值:	250
语法:	*Object*.ColorPriorityRange1 *[= integer or priority]*

属性:	**ColorPriorityRange2**
用途:	设置要显示的报警的优先级范围之边界。此属性的值必须大于 ColorPriorityRange1 且小于 ColorPriorityRange3。
类型:	Integer
缺省值:	500
语法:	*Object*.ColorPriorityRange2 *[= integer or priority]*

属性:	**ColorPriorityRange3**
用途:	设置要显示的报警的优先级范围之边界。此属性的值必须大于 ColorPriorityRange2 且小于 999。
类型:	Integer
缺省值:	750
语法:	*Object*.ColorPriorityRange3 *[= integer or priority]*

属性:	**EventForeColor**	
用途:	返回或设置事件报警的前景颜色。此颜色应用于控件中所显示的状态为 EVT_EVT 的记录。	
类型:	整型	
缺省值:	品红色	
语法:	*Object*.EventForeColor *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**EventBackColor**	
用途：	返回或设置事件报警的背景颜色。此颜色应用于控件中所显示的状态为 EVT_EVT 的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.EventBackColor *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**FilterFavoritesFile**
用途：	字符串属性。设置 / 返回过滤器收藏夹文件。此文件有 " 过滤器收藏夹 " 对话框用于读取 / 写入过滤器收藏夹。
类型：	Discrete
缺省值：	`Null`
语法：	*Object*.FilterFavoritesFile *[= Discrete]*

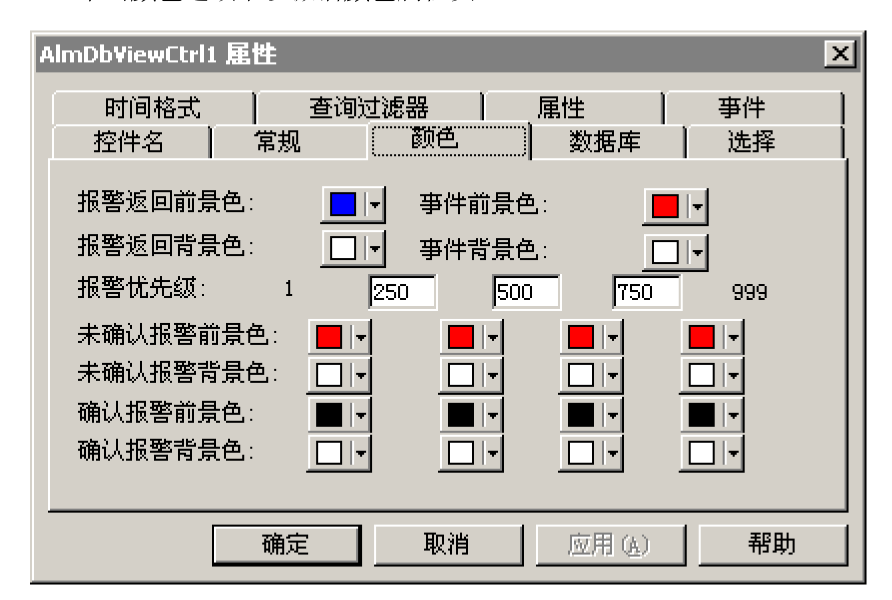
属性：	**FilterMenu**		
用途：	返回或设置一个值，该值确定上下文相关菜单中是否显示 " 过滤器 " 菜单项。		
类型：	Discrete		
缺省值：	`True`		
语法：	*Object*.FilterMenu *[= Discrete]*		
		True	（缺省值）显示 " 过滤器 " 菜单项。
		False	不显示 " 过滤器 " 菜单项。

属性：	**FilterName**
用途：	返回当前过滤器的名称 （若存在）。
类型：	String (Read Only)
缺省值：	`Null`
语法：	*Object*.FilterName *[= String]*

配置报警显示颜色

要配置报警显示颜色

1. 双击 Alarm DB View 控件，或右击它，然后单击**属性**。此时会出现 **AlarmDBViewCtrlX** 对话框，并显示**控件名**属性页。
2. 单击**颜色**选项卡以激活**颜色**属性页。



3. 单击每个颜色框以打开 InTouch 调色板。在调色板上为下列各项单击您要使用的颜色：

   报警返回前景色
   设置报警返回显示的前景颜色。

   报警返回背景色

   设置报警返回显示的背景颜色。

   事件前景

   设置事件显示的前景颜色。

   事件背景色
   设置事件显示的背景颜色。

4. 在**报警优先级**框中，输入用于报警显示的断点值。您可以指定多个断点值，这样报警便会根据 “报警优先级” 显示为不同的颜色。预先确定的最小与最大报警优先级值分别是 1 与 999。

颜色优先级	范围
1 到 ColorPriorityRange1	优先级在范围 1 到第一个报警优先级框中的断点值之间的报警。缺省值设置为 250。
ColorPriorityRange1 到 ColorPriorityRange2	优先级范围在第一个报警优先级框中的断点值与第二个框中的断点值之间的报警。缺省值是第二个报警优先级框中的值设置为 250，第三个框中的值设置为 500。
ColorPriorityRange2 到 ColorPriorityRange3	优先级范围在第二个报警优先级框中的断点值与第三个框中的断点值之间的报警。缺省值是第一个报警优先级框中的值设置为 500，第二个框中的值设置为 700。
ColorPriorityRange3 到 999	优先级在第三个报警优先级框中的断点值与 999 之间的报警。缺省值是第三个报警优先级框中设置为 750。最大值是 999。

5. 为 “未确认报警前景色” 的每个 ColorPriorityRange （颜色优先级范围）设置颜色。单击每个颜色框以打开 InTouch 调色板。
6. 为 “未确认报警背景色” 的每个 ColorPriorityRange （颜色优先级范围）设置颜色。
7. 为 “确认报警前景色” 的每个 ColorPriorityRange （颜色优先级范围）设置颜色。
8. 为 “未确认报警背景色” 的每个 ColorPriorityRange （颜色优先级范围）设置颜色。
9. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

属性：	**UnAckAlmForeColorRange1**	
用途：	返回或设置未确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 1 到 ColorPriorityRange1 之间的记录。	
类型：	整型	
缺省值：	红色	
语法：	*Object*.UnAckAlmForeColorRange1 *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmForeColorRange2**	
用途：	返回或设置未确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 ColorPriorityRange1 到 ColorPriorityRange2 之间的记录。	
类型：	整型	
缺省值：	红色	
语法：	*Object*.UnAckAlmForeColorRange2 *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmForeColorRange3**	
用途：	返回或设置未确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 ColorPriorityRange2 到 ColorPriorityRange3 之间的记录。	
类型：	整型	
缺省值：	红色	
语法：	*Object*.UnAckAlmForeColorRange3 *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmForeColorRange4**	
用途：	返回或设置未确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 ColorPriorityRang 到 999 之间的记录。	
类型：	整型	
缺省值：	红色	
语法：	*Object*.UnAckAlmForeColorRange4 *[= color]*	
	color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmBackColorRange1**	
用途：	返回或设置未确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 1 到 ColorPriorityRange1 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.UnAckAlmBackColorRange1 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmBackColorRange2**	
用途：	返回或设置未确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 ColorPriorityRange1 到 ColorPriorityRange2 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.UnAckAlmBackColorRange2 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmBackColorRange3**	
用途：	返回或设置未确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 ColorPriorityRange2 到 ColorPriorityRange3 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.UnAckAlmBackColorRange3 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**UnAckAlmBackColorRange4**	
用途：	返回或设置未确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 UNACK_ALM 且优先级在范围 ColorPriorityRang 到 999 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.UnAckAlmBackColorRange4 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**AckAlmBackColorRange1**	
用途：	返回或设置已确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 1 到 ColorPriorityRange1 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.AckAlmBackColorRange1 *[= color]*	
	color	一个值，或一个常数，用于确定背景的颜色。

属性：	**AckAlmBackColorRange2**	
用途：	返回或设置已确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 ColorPriorityRange1 到 ColorPriorityRange2 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.AckAlmBackColorRange2 *[= color]*	
	color	一个值，或一个常数，用于确定背景的颜色。

属性：	**AckAlmBackColorRange3**	
用途：	返回或设置已确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 ColorPriorityRange2 到 ColorPriorityRange3 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.AckAlmBackColorRange3 *[= color]*	
	color	一个值，或一个常数，用于确定背景的颜色。

属性：	**AckAlmBackColorRange4**	
用途：	返回或设置已确认的报警的背景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 ColorPriorityRange3 到 999 之间的记录。	
类型：	整型	
缺省值：	窗口背景颜色	
语法：	*Object*.AckAlmBackColorRange4 *[= color]*	
	color	一个值，或一个常数，用于确定背景的颜色。

属性：	**AckAlmForeColorRange1**	
用途：	返回或设置已确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 1 到 ColorPriorityRange1 之间的记录。	
类型：	整型	
缺省值：	蓝色	
语法：	*Object*.AckAlmForeColorRange1 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**AckAlmForeColorRange2**	
用途：	返回或设置已确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 ColorPriorityRange1 到 ColorPriorityRange2 之间的记录。	
类型：	整型	
缺省值：	蓝色	
语法：	*Object*.AckAlmForeColorRange2 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

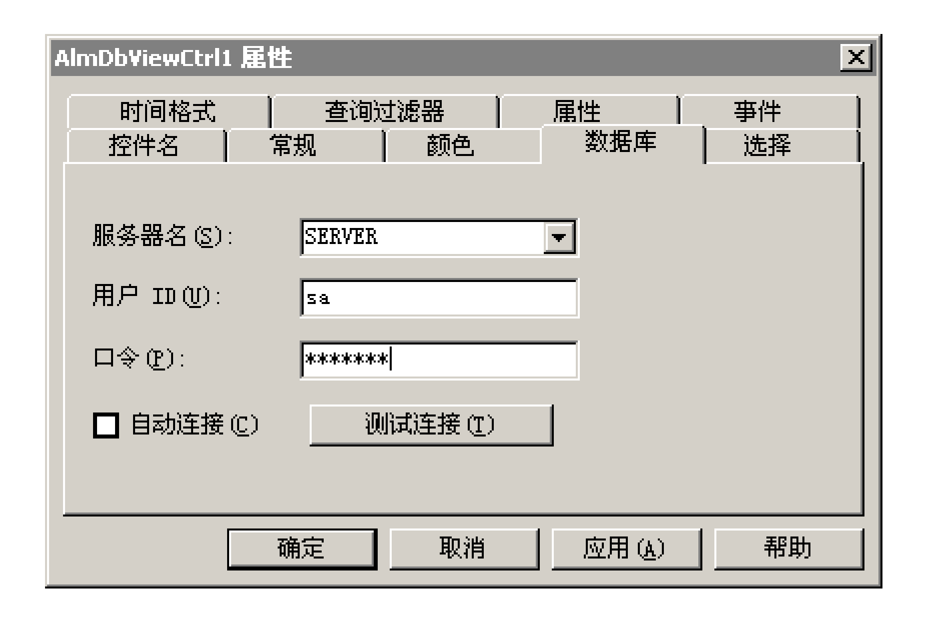
属性：	**AckAlmForeColorRange3**	
用途：	返回或设置已确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 ColorPriorityRange2 到 ColorPriorityRange3 之间的记录。	
类型：	整型	
缺省值：	蓝色	
语法：	*Object*.AckAlmForeColorRange3 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

属性：	**AckAlmForeColorRange4**	
用途：	返回或设置已确认的报警的前景颜色。此颜色会应用于控件中所显示的状态为 ACK_ALM 且优先级在范围 ColorPriorityRange3 到 999 之间的记录。	
类型：	整型	
缺省值：	蓝色	
语法：	*Object*.AckAlmForeColorRange4 *[= color]*	
	Color	一个值，或一个常数，用于确定指定对象的颜色。

数据库页

“数据库”选项卡用于设置服务器名及其相关属性，使控件可以在需要时连接到数据库。

备注 使用对报警数据库有适当访问权限但并非系统管理员帐户的用户 ID 是要一种很好的做法。



服务器名

服务器名下拉列表框由网络中可用 SQL Server 与 MSDE Server 的名称列表组成。用户可以编辑此列表框。此图形用户界面功能同以下属性有关：

属性：	**ServerName**
用途：	返回或设置服务器名，控件会连接到该服务器上以获取数据。
类型：	消息
语法：	*Object*.ServerName *[= text]*

用户

用户字段用于输入数据库服务器的用户名，以建立所需的连接。此图形用户界面功能同以下属性有关：

属性：	**UserID**	
用途：	返回或设置用户 ID，在控件连接到 SQL Server 以获取数据时会用到它。	
类型：	消息	
语法：	*Object*.UserID *[= text]*	
	text	表示用户 ID 的字符串表达式。

口令

口令字段用于输入数据库服务器的用户名所对应的口令，以建立所需的连接。此图形用户界面功能同以下属性有关：

属性：	**Password**	
用途：	返回或设置口令，在控件连接到 SQL Server 以获取数据时会用到它。	
类型：	消息	
语法：	*Object*.Password *[= text]*	
	text	表示口令的字符串表达式。

测试连接

测试连接按钮用于在设计时使用给定的服务器名、用户名及其口令来测试与 WWALMDB 数据库的连接。

自动连接复选框

在缺省情况下，**自动连接**复选框是禁用的。填写服务器名与用户 ID 字段之后，此复选框会变成启用状态。此复选框可用于在运行时启用或禁用控件自动连接服务器数据库的功能。此图形用户界面功能同以下属性有关：

属性：	**AutoConnect**		
用途：	返回或设置一个值，该值确定在控件进入运行模式后是否立即连接数据库。		
类型：	整型		
缺省值：	`False`		
语法：	*Object*.AutoConnect *[= Integer]*		
	整型	整型表达式，指定控件是否如 “设置”部分所述进入运行模式后立即连接数据库。	
		True	连接数据库。
		False	（缺省值）不连接数据库。

备注 如果在设计时此属性被设置为 `False`，则在运行模式下该控件不连接数据库。此时，若要连接数据库，则必须显式调用 "Connect" 方法。

选择选项卡

选择选项卡提供了需要根据时间进行检索的记录集的有关详细信息。它还给出了可显示的最大记录数、是否显示日期与时间，以及日期与时间的显示格式。



使用指定时间复选框

此复选框用于启用或禁用用户是否可以为要检索的记录输入 “开始时间” 与 “结束时间” 以及是否基于**持续时间**属性来计算它们的功能。此图形用户界面功能同以下属性有关：

属性：	**SpecificTime**		
用途：	返回或设置一个值，该值确定控件是使用 “开始时间” 与 “结束时间” 属性，还是根据 “持续时间” 属性的值来计算开始时间与结束时间。		
类型：	离散		
缺省值：	`False`		
语法：	*Object*.SpecificTime *[= Discrete]*		
		True	使用 “开始时间” 与 “结束时间” 属性。
		False	（缺省值）根据 “持续时间” 属性来计算 “开始时间” 与 “结束时间”。

持续时间

取消选择**使用指定时间**复选框之后，**持续时间**下拉列表框便会启用。该列表框提供了一个相对于当前时间的预定义时间间隔列表。此图形用户界面功能同以下属性有关：

属性：	**Duration**	
用途：	返回或设置持续时间，该时间由控件用于设置 “开始时间” 与 “结束时间”。	
类型：	消息	
缺省值：	“最近 1 小时”	
语法：	*Object*.Duration *[= text]*	
	text	表示持续时间的字符串表达式。

备注 此属性必须是下列字符串之一：
最近 1 分钟
最近 5 分钟
最近 15 分钟
最近 30 分钟
最近 1 小时
最近 2 小时
最近 4 小时
最近 8 小时
最近 12 小时
最近 1 天
最近 2 天
最近 3 天
最近 1 星期
最近 2 星期
最近 30 天
最近 90 天

开始时间

仅当选中**使用指定时间**复选框时，此日期 - 时间控件才会启用。此属性用于设置一定时间间隔中要检索的报警记录的开始时间。此图形用户界面功能同以下属性有关：

属性：	**StartTime**	
用途：	返回或设置开始时间。	
类型：	消息	
语法：	*Object.*StartTime *[= text]*	
	text	表示 “开始时间”的字符串表达式。

备注 所返回的字符串采用的格式是在**日期与时间**属性中设置的。要对值进行设置，此字符串必须采用 MM/DD/YYYY HH:MM:SS 格式。此属性可以处理自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 的任意时区中的日期。

结束时间

仅当选中**使用指定时间**复选框时，此日期 - 时间控件才会启用。此控件用于设置一定时间间隔中要检索的报警记录的结束时间。此图形用户界面功能同以下属性有关：

属性：	**EndTime**	
用途：	返回或设置结束时间。	
类型：	消息	
语法：	*Object*.EndTime *[= text]*	
	text	表示结束时间的字符串表达式。

备注 所返回的字符串采用的格式是在**日期与时间**属性中设置的。要对值进行设置，此字符串必须采用 MM/DD/YYYY HH:MM:SS 格式。此属性可以处理自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 的任意时区中的日期。

最大记录数

最大记录数字段用于编辑一次可以在控件中显示的记录数。此图形用户界面功能同以下属性有关：

属性：	**MaxRecords**	
用途：	返回或设置一个值，该值用于指定在给定时间要获取的最大记录数。	
类型：	整型	
缺省值：	100	
语法：	*Object*.MaxRecords *[=integer]*	
	整数	整型表达式，指定在给定时间要获取的记录数。

备注 最大记录数的范围是从 1 到 1000。为确保取得最佳性能，应根据实际需要将此值设置得尽可能小。

持续时间列

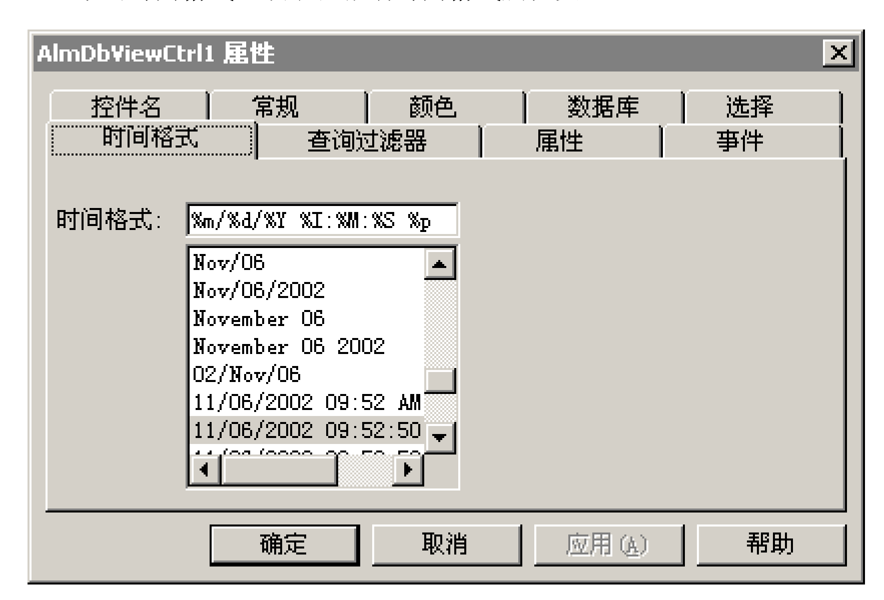
“持续时间列”显示 “未确认 / 报警持续时间”的毫秒数。

- 选择**未确认持续时间**可显示最近的报警状态变化 （ALM 或子状态）与 ACK 时间的时间，如果有的话。
- 选择**报警持续时间**可显示报警最初发生时与它返回正常状态时之间所经过的时间量。

时间格式选项卡

要配置 Alarm DBView 控件时间格式

1. 双击 Alarm DBView ActiveX 控件，或右击它，然后单击**属性**。此时会出现 **AlarmDBViewCtrlX 属性**对话框，并显示**控件名**属性页。
2. 单击**时间格式**选项卡以激活**时间格式**属性页。



滚动 “时间格式”选项以选择所需的时间格式。可用的选项包括：:

字符串字符	描述
d	两位数字的日期 - 09
b	3 个字母的月份缩写 - Aug
Y	四位数字的年份 - 2002
m	两位数字的月份 - 08
/	日期分隔符，如 08/09 中的那样
y	两位数字的年份 - 02
#x	完整的日期与星期 - Friday, August 09, 2002
B	完整的月份名称 - August
-	日期分隔符，如 08-09 中的那样
.	日期分隔符，如 08-09 中的那样
,	日期分隔符号，如 Aug 09, 2002 中的那样
H	24 小时制表示的时间 - 16:00
:	时间分隔符，如 4:41 中的那样
M	分钟 00:41

字符串字符	描述
P	PM
S	秒钟 - 16:41:07
s	小数秒 - 16:41:07.390
I	要求指定 AM/PM 的 12 小时制时间表示法 - 04:41 PM

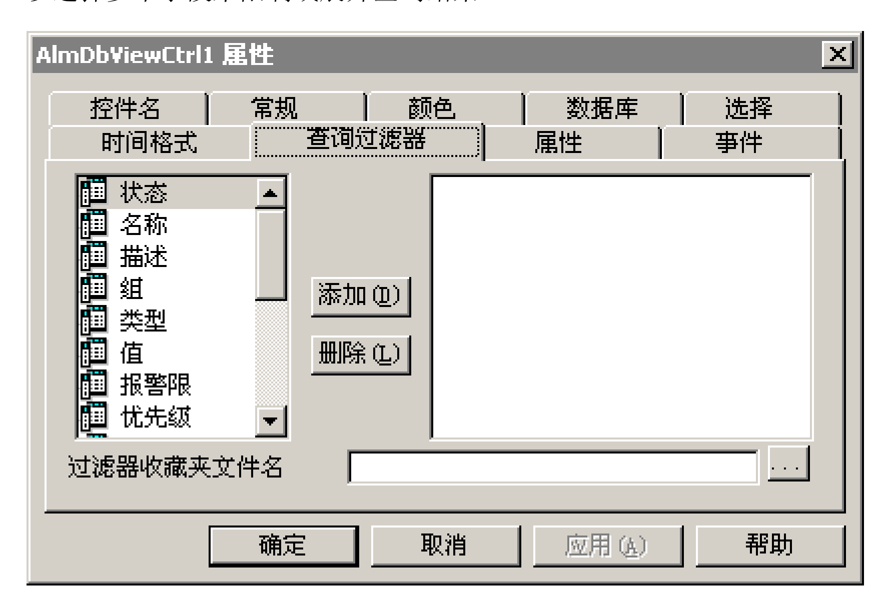
下面是一些时间格式字符串示例：

时间格式字符串	显示
%d %b	09 Aug
%m/%d/%Y	08/09/2002
%#x	Friday, August 09, 2002
%Y-%m-%d	2002-08-09
%m/%d/%Y %H:%M %p	08/09/2002 16:56 PM
%m/%d/%Y %H:%M:%s %p	08/09/2002 16:56:38.07
%I:%M %p	04:56 PM

3. 单击**应用**。您可以继续配置下一个属性，或单击**确定**以退出属性页。

查询过滤器选项卡

查询过滤器选项卡可供您通过将列选作过滤器来定义要在查询结果中包含哪些记录。例如，您可以按记录日期或报警状态来选择过滤器。您可以选择多个字段来限制或展开查询结果。



选择查询过滤器

所有的过滤器列均显示在左侧窗格中。已被选作过滤器的列则显示在右侧窗格中。

要选择查询过滤器

- 在左侧窗格中双击某个列，或者选择某个列然后单击**添加**而将他移到过滤器窗格中。



关于过滤器窗格

过滤器（右侧）窗格包含作为查询的一部分的各个列的列表。定义多个列时，这些列会通过使用“与”或“或”加以合并。选择“与”运算符时，仅返回满足显示的列的所有值的记录。选择“或”时，返回的记录只要满足这些列中任一个的值即可。缺省情况下，多个列总是会组合为一个“与”表达式。若要使用 And/Or 语句，则必须将相应的列组合到一起。在过滤器窗格中的某一项上，只能创建一个过滤器表达式。如果需要多个表达式，则必须将该项目再添加到过滤器窗格中。若要从过滤器窗口中删除某个列，请单击要删除的列，然后单击**删除**。

报警列

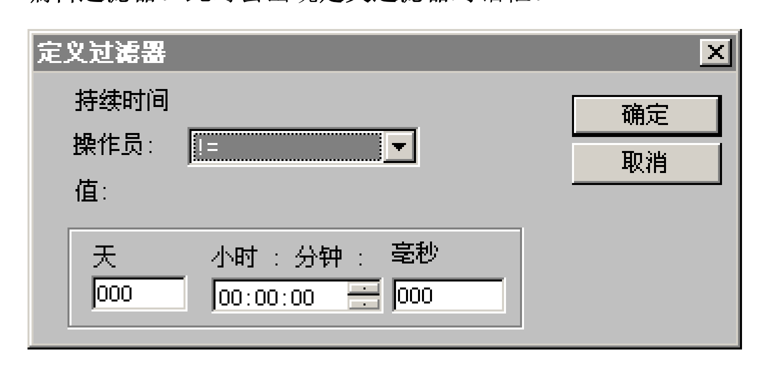
下面介绍**查询过滤器**选项卡左侧窗格中的报警列：

列	描述
状态	按报警状态过滤查询。您可以从下拉菜单中选择所需的报警状态。
名称	按报警名过滤查询。
描述	按报警描述过滤查询。
组	按报警组过滤查询。
类型	按报警类型过滤查询。

列	描述
值	按报警值过滤查询。“值”列中的值显示为数值。“查询过滤器”中这些值的比较是按字符串的比较方式来完成的。
报警限	按报警限过滤查询。“报警限”列中的值显示为数值。“查询过滤器”中这些值的比较是按字符串的比较方式来完成的。
优先级	按报警优先级过滤查询。
类别	按报警类过滤查询。
供应器	按报警供应器过滤查询。
操作员	按操作员过滤查询。
域名	按报警域名过滤查询。
用户全名	按用户的全名过滤查询。
持续时间	按未确认持续时间过滤查询。
用户 1	按报警用户定义的值 1 过滤查询。
用户 2	按报警用户定义的值 2 过滤查询。
用户 3	按报警用户定义的值 3 过滤查询。

要给列定义过滤器

1. 双击过滤器窗口中的列，或右击选定的列，然后单击快捷菜单中的**编辑过滤器**。此时会出现**定义过滤器**对话框。



2. 单击**运算符**下列箭头以选择所需的运算符。可用运算符列表中包含的运算符有：等于 =、不等于 !=、小于等于 <=、大于等于 >=、小于 <、大于 >、**相似** 以及**不相似**。

3. 在**值**框中输入一个值，然后单击**确定**。

备注 “值”框不会接受选定的查询无法处理的数据。给数值列名使用“相似”与“不相似”运算符时，“值”框可以接受下列字符，并将它们当作 SQL 服务器的通配符：
% - 0 个或更多字符组成的任何字符串。
_ - 任意单个字符。
[ ] - 不在指定范围 ([^a-f]) 或集合 ([^abcdef]) 中的任意单个字符。
[^] - 不在指定范围 ([^a-f]) 或集合 ([^abcdef]) 中的任意单个字符。
“持续时间”、“用户 1”、“用户 2”、“状态”、“优先级”列之外的所有列的“值”文本框均可接受所有的字母数字字符。
对于“优先级”列，“值”文本框仅接受从 1 到 999 之间的整数值。
“持续时间”的“日期”文本框的最大值是 99999，“毫秒”文本框的最大值是 999。负数不允许使用。
“用户 1”与“用户 2”列只能接受数字，既可以是负数、整数，也可以是分数。

状态列的值

将**状态**列添加到过滤器查询时，您可以从“定义过滤器”对话框的“值”下拉菜单中给它指定值。下表介绍这些可以使用的值：

值	描述
ACK	生成所有系统 ACK 的查询。
ACK_ALM	生成所有已确认报警的查询。
UNACK_ALM	生成所有未确认报警的查询。
ACK_RTN	生成所有 ACK_RTN 的查询。
UNACK_RTN	生成所有 UNACK_RTN 的查询。
所有未确认记录	生成所有 UNACK 记录的查询。
所有确认记录	生成所有 ACK 记录的查询。
所有报警记录	生成所有报警记录的查询。
所有 RTN 记录	生成所有 RTN 记录的查询。

备注 扩展的摘要报警模式中的标记用于创建报警，而该报警在主报警被确认后又回到正常状态，会创建两个记录。第一个记录是 ACK_RTN 记录，因为此时新报警已经回到正常状态。第二个记录是 ACK，它对应于确认主报警。先前 ACK_ALM 的实现方式已经更改为 ACK。

要组合多个列

1. 右击某个列，然后从快捷菜单中选择**组合**。
2. 将一个列拖放到另一个列上。

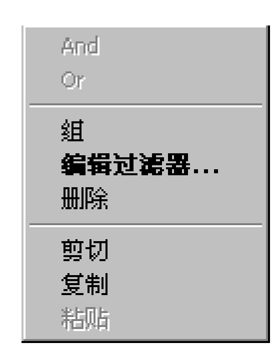
备注 “与”与“或”运算符是父节点。每个父节点下面选定的列是子节点。您无法将父节点拖放到子节点上。

缺省情况下，组合到一起的列会使用 “与”运算符。若要将它更改为 “或”，请右击运算符，然后从快捷菜单中选择适当的运算符。

备注 如果设计时尚未定义过滤器，则 “过滤器收藏夹”对话框中会显示一条叫 “缺省过滤器”的记录。如果您尚未在**查询过滤器**选项卡中定义任何过滤器，则**过滤器收藏夹**对话框中的 “缺省过滤器”在运行时会查询所有记录。

关于快捷菜单

过滤器窗格中提供了一个快捷菜单。



右击 “过滤器”窗格中的某个列即可打开该快捷菜单。菜单选项包括：

菜单选项	功能
与	将 “或”节点更改为 “与”节点。除了 “或”以外，此选项对于所有其它的菜单项均是禁用的。
或	将 “与”节点更改为 “或”节点。除了 “与”以外，此选项对于所有其它的菜单项均是禁用的。
组合	创建一个项目的拖动映像，并开始拖放操作。
编辑过滤器	弹出**定义过滤器**对话框。
剪切	剪切所选的过滤器。
复制	复制所选的过滤器。
粘贴	粘贴剪切或复制的过滤器。
删除	删除过滤器表达式。

复制或移动查询过滤器

如果有多个 Alarm DBView 的实例，并且您想要给多个实例使用相同的过滤器，则您可以从一个实例中复制 （或剪切）定义好的过滤器，然后将它们粘贴到另一个实例中。

要将过滤器从 Alarm DB View 的一个实例复制到另一个实例

1. 在 Alarm DBView 的第一个实例中定义所需的过滤器。
2. 右击过滤器，然后单击**复制** （要移动过滤器，请单击**剪切**）。

3. 单击**确定**或**取消**，以关闭 Alarm DBView 的第一个实例。
4. 打开 Alarm DBView 的下一个实例，然后单击**查询过滤器**选项卡。
5. 将指针放到右侧窗格，右击所选的过滤器，然后单击**粘贴**。

备注 要激活快捷菜单，您至少要将一个过滤器属性添加到右侧窗格。

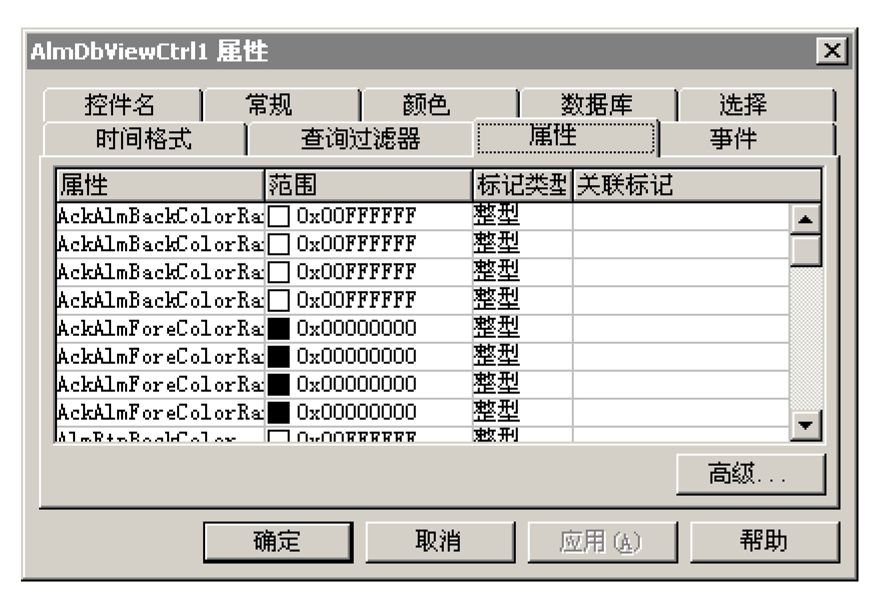
加载查询过滤器收藏夹

要加载查询过滤器收藏夹

1. 在**查询过滤器**选项卡中，在**过滤器收藏夹名**文本框中，输入网络路径与文件名，或单击按钮以浏览文件。
2. 如果您选择浏览文件，请从网络上选择，然后单击**打开**。
3. 单击**应用**。

属性页

属性选项卡用于将标记名指定给此控件的各种可用属性。



通用属性

属性：	**RowCount**
用途：	返回控件中当前显示的记录数。此属性为**只读**属性。
类型：	整型
语法：	*Object*.RowCount
示例：	***tagname* = #AlmDbView1.RowCount;** （其中，控件的名称是 AlmDbView1，标记名定义为一个整型标记名）。

属性：	**TotalRowCount**
用途：	返回当前查询的总记录数。此属性为**只读**属性。
类型：	整型
语法：	*Object*.TotalRowCount
示例：	***tagname* = #AlmDbView1.TotalRowCount;** （其中，控件的名称是 AlmDbView1，标记名定义为一个整型标记名）。

备注 RowCount 是当前查询所返回的行数，通常与 MaxRecords 属性相同，但检索到的记录数小于 MaxRecords 属性时，二者将不相同。例如，如果符合特定查询标准的记录数为 950，而最大记录数为 100，则最后一页将显示 50 条记录，因此行数会是 50。在相同的示例中，TotalRowCount 始终为 950。

方法与事件

属性：	**ConnectStatus**
用途：	返回连接的状态。此属性为只读属性。
类型：	消息
语法：	*Object*.ConnectStatus
示例：	`Tagname = #AlmDbView1.ConnectStatus;` （其中，控件的名称是 AlmDbView1，标记名定义为“消息”型标记名）。
返回值	
已连接	控件已连接到数据库。
未连接	控件未连接到数据库。
进行中	控件正在连接数据库。

方法：	**Connect**
用途：	将控件连接到数据库，如果连接成功，则显示范围从 1 到 MaxRecords 之间的记录集。
语法：	*Object*.Connect
示例：	`#AlmDbView1.Connect();` （其中，控件的名称是 AlmDbView1）。

方法：	**Disconnect**
用途：	断开控件与数据库的连接。
语法：	*Object*.Disconnect
示例：	`#AlmDbView1.Disconnect();` （其中，控件的名称是 AlmDbView1）

方法：	**Refresh**
用途：	从数据库刷新控件，如果连接成功，则显示范围从 1 到 MaxRecords 之间的记录集。
语法：	*Object*.Refresh
示例：	**`#AlmDbView1.Refresh();`** （其中，控件的名称是 AlmDbView1）

方法：	**Reset**
用途：	将所有的列重置为设计时保存的设置。
语法：	*Object*.Reset
示例：	**`#AlmDbView1.Reset();`** （其中，控件的名称是 AlmDbView1）。

方法：	**ShowSort**
用途：	在启用 "SortMenu" 属性的情况下，显示 “辅助排序”对话框。
语法：	*Object*.ShowSort
示例：	**`#AlmDbView1.ShowSort();`** （其中，控件的名称是 AlmDbView1）。

方法：	**ShowContext**
用途：	在启用 "RefreshMenu" 或 "ResetMenu" 或 "SortMenu" 等任一属性的情况下，显示上下文相关菜单。
语法：	*Object*.ShowContext
示例：	**`#AlmDbView1.ShowContext();`** （其中，控件的名称是 AlmDbView1）。

方法：	ShowFilter
用途：	显示 “过滤器收藏夹” 对话框。
语法：	*Object*.ShowFilter
示例：	**#AlmDbView1.ShowFilter();** （其中，控件的名称是 AlmDbView1）。

方法：	GetItem	
用途：	以字符串的形式返回指定的行和列的数据。	
语法：	*Object*.GetItem*(Integer, message)*	
	整型	表示控件中特定某行的整型表达式。
	消息	表示控件中的列名的字符串表达式。
示例：	***tagname*** **= #AlmDbView1.GetItem(1, "Group");** （其中，控件的名称是 AlmDbView1，标记名定义为 “消息” 型标记名）。	

方法：	GetSelectedItem	
用途：	以字符串的形式返回指定的行和列的数据	
语法：	*Object*.GetSelectedItem*(message)*	
	消息	表示控件中的列名的字符串表达式。
示例：	***tagname*** **= #AlmDbView1.GetSelectedItem("State");** （其中，控件的名称是 AlmDbView1，标记名定义为 “消息” 型标记名）。	

方法：	GetNext
用途：	从数据库检索下一组记录 （如果有的话）。
语法：	*Object*.GetNext
示例：	**#AlmDbView1.GetNext();** （其中，控件的名称是 AlmDbView1）。

方法：	**GetPrevious**
用途：	从数据库检索上一组记录 （如果有的话）。
语法：	*Object.*GetPrevious
示例：	**`#AlmDbView1.GetPrevious();`** （其中，控件的名称是 AlmDbView1）。

方法：	**GetLastError**	
用途：	返回无提示模式下的最后一条错误消息。	
语法：	Object. GetLastError()	
示例：	读取：	**`Tagname = #AlmDbView1.GetLastError();`**
	其中，控件的名称是 AlmDbView1，标记名定义为变量或字符串。	

方法：	**AboutBox**
用途：	显示 “关于”对话框。
语法：	*Object.*AboutBox
示例：	**`#AlmDbView1.AboutBox();`**
	其中，控件的名称是 AlmDbView1。

事件页

事件选项卡用于将某些脚本函数指定给可以在此控件上发生的各种事件。



事件：	**Click**	
用途：	用户在控件上单击并释放鼠标左键后发生。	
语法：	*tagname = #Thisevent*.clicknRow	
	nRow	整型表达式，反映发生的事件所在的行。
	#Thisevent	关键字

备注 从 OnStartup 事件调用此方法时， AlarmDB View 控件会忽略 UI 方法，因为此控件尚不是可见的。下面列出了一些 UI 方法，如果它们是从 OnStartup 事件中调用的，则会被忽略掉：
"ShowSort"、 "ShowContext"、 "GetSelectedItem"、 "GetNext"、 "GetPrevious" 以及 "AboutBox"。

事件：	**DoubleClick**	
用途：	用户在控件上双击并释放鼠标左键时发生。	
语法：	*Tagname = #ThisEvent*.DoubleClicknRow	
	nRow	整型表达式，反映发生的事件所在的行。
	#Thisevent	关键字

Click 与 DoubleClick 事件均是零基的。给用户发布 Click 与 / 或 DoubleClick 事件时，显示中的行计数是从 0 开始的。

第 11 章

报警实用程序

第 9 章 “报警 / 事件”所述的 InTouch 通知系统支持显示、记录与打印过程报警与系统事件。本章介绍 “报警打印程序实用程序”与 Alarm DB Logger 实用程序的运作。

目录

报警打印程序实用程序

“报警打印程序”实用程序提供了一个集中化管理工具，可供用户使用这个独立的工具跨越多个节点打印出分布式报警信息。通过使用 “报警打印程序”实用程序，您可以使用专用的行式打印机，逐个事件地打印报警状态的变化。您可以定义不同的打印选项，并将这些定义保存在 “报警打印程序”实用程序配置文件 (*.ALC) 中，供日后检索使用。

报警状态的每次改变均会生成一个事件，可以为诊断问题提供有益的信息。“分布式报警系统”可以配置成发生特定事件时在行式打印机上将它们打印出来。通常，在发生灾难性系统故障和 （或）掉电时，客户希望将此打印件视为最后的恢复手段。一般而言，这指的是通过串行端口或并行端口打印到点阵打印机上。Windows 网络打印机与激光打印机通常不适合此项功能，因为它们在实际打印之前会将整个页面缓冲到内存中 － 这意味着在系统崩溃或掉电时还是可能会丢失信息。

备注 “报警打印程序”实用程序只查询历史数据，而不查询摘要数据。

报警打印日期 / 时间标签

“报警打印程序”的缺省日期格式为：

DD MM YYYY

其中， **DD** 指日期， **MMM** 指月份， **YYYY** 指年份。

“报警打印程序”的缺省时间格式为：

HH:MM:SS.SSS

其中， **HH** 是时间， **MM** 是分钟， **SS** 是秒钟， **SSS** 是毫秒。

备注 无论计算机的时区设置如何，报警记录的时间字段始终为 **GMT** 格式。

您可以在**配置设置**对话框的**消息**属性页中更改日期和 （或）时间的缺省格式。

如需有关详细信息，请参阅 “配置报警打印程序”。

使用一个或多个报警打印程序

您可以同时运行 “报警打印程序”的多个实例 (AlmPrt.exe)。“报警打印程序”的每个实例均必须配置为打印到不同的打印机，并且必须使用自己单独的报警查询进行配置。

提示 您可以配置 “报警打印程序”的不同实例，以打印优先级在特定范围内的报警。例如，某个 “报警打印程序”实例可以只打印高优先级报警，而另一个实例则只打印低优先级报警。与此类似，您可以使用 “报警打印程序”的一个实例打印来自工厂某个区域的报警，而使用另外一个实例打印另一不同区域的报警。

使用报警打印程序

初次启动 “报警打印程序”时，会出现一个**报警打印程序**窗口，显示 “报警打印程序”的缺省配置。

备注 如果运行时在命令提示符下打开它，或双击其 *.ALC 文件名，则可能会显示一个特定的报警配置文件。



要创建新的报警打印程序配置文件

1. 在**文件**菜单上，选择**新建**。“报警打印程序”的缺省**报警状态**为“全部”，**优先级**为 1-999，**打印端口** < 无 >。
2. 单击**配置**菜单。此时会出现**配置设置**对话框。输入所需的设置。
3. 在“文件”菜单上，选择**保存**。

要编辑现有的报警打印程序配置文件

1. 在**文件**菜单上，选择**打开**。
2. 选择要编辑的“报警打印程序”配置文件。
3. 要保存更改，请在**文件**菜单上选择**保存**。
4. 或者是，在**文件**菜单上选择**另存为**，将更改保存到一个新文件，而不更改现有的文件。

如需有关“报警打印程序”配置的详细信息，请参阅“配置报警打印程序”。

报警打印程序工具栏



“报警打印程序”工具栏上的工具可供您快速访问所有的 “报警打印程序”命令。

下面标出每个工具的插图并附上相应的说明：

工具	描述
	创建新的 “报警打印程序”配置文件。
	打开现有的 “报警打印程序”配置文件。
	将当前的 “报警打印程序”配置文件保存到磁盘。
	配置 “报警打印程序”。
	启动 / 停止查询。
	启用 / 禁用报警打印功能。

配置报警打印程序

“报警打印程序”配置保存在 “报警打印程序配置文件” (*.alc) 中。可以保存的 “配置”文件数上没有任何限制。对于正在运行的 “报警打印程序”的每个实例，“报警打印程序”只能使用一个配置文件。

要配置报警打印程序查询属性

1. 选择**配置**，或单击 “报警打印程序”工具栏上的 “配置报警查询”工具。

2. 此时会出现**配置设置**对话框。

3. 在**从优先级**字段中，输入报警的最高优先级值 （1 到 999）。
4. 在**到优先级**字段中，输入报警的最低优先级值 （1 到 999）。

备注 InTouch 中配置的每个报警都有一个关联的优先级值。此值表示报警的严重程度，范围从 1 到 999，其中 1 表示最为严重。

如需有关报警优先级详细信息，请参阅 “报警优先级”。

5. 单击**报警状态**箭头，从列表中选择您要使用的报警状态。
6. 在**报警查询**输入框中，输入您要执行的 InTouch 报警查询组。您可以指定一个或多个 “报警供应器”与报警组，方法同处理 “分布式报警显示”的报警查询一样。如需有关详细信息，请参阅 “配置分布式报警显示”。

要配置报警记录属性

1. 在**输出**菜单上，选择**配置**，或单击 “报警打印程序”工具栏上的 “配置报警查询”工具。
2. 此时会出现**配置设置**对话框。

3. 选择**消息**选项卡以激活**消息**属性页：

提示 您选择的每个选项在生成的打印件中将显示为一个单独的字段。设置为特定字符长度的字段将按照该长度限制进行截断处理。

如果右击任何报警配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

备注 大多数打印机每行可打印 80 或 132 个字符。

4. 要打印报警日期，请选择**日期**选项，然后单击箭头选择日期格式。可用的格式包括：

选择	格式	选择	格式
DD MMM	28 Feb	**MM/DD**	02/28
DD MM YYYY	28 Feb 1997	**MM/DD/YY**	02/28/97
DD/MM	28/02	**MMM DD**	Feb 28
DD/MM/YY	28/02/97	**MMM DD YYYY**	Feb 28 1997

5. 要打印报警时间，请选择**时间**选项，然后单击箭头选择时间格式。可用的格式包括：

选择	描述
AP	选择 AM/PM 格式。例如，下午三点钟显示为 3:00 PM。非此格式的时间缺省使用 24 小时军用时间格式。例如，下午三点钟显示为 15:00。
HH	打印发生报警 / 事件时的小时值。
MM	打印发生报警 / 事件时的分钟值。
SS	打印发生报警 / 事件时的秒钟值。
SSS	打印发生报警 / 事件时的毫秒值。

6. 在**时间**下面的 “排序顺序” 框内，选择报警记录中的报警排序顺序。这里总共有三个选项：

选择	描述
OAT	原始报警时间 - 即报警发生的日期 / 时间标签。
LCT	上次更改时间 - 即报警实例最近一次改变状态的日期 / 时间标签：包括报警发生、子状态改变、恢复正常或确认。
LCT，但确认后用 OAT	“上次更改时间”，但确认后用 “原始报警时间” - 也就是，撤消确认报警时使用 LCT，但在确认报警后使用 OAT。

7. 要打印报警状态，请选择**报警状态 (UnAck, Ack)** 选项。
8. 要打印报警类，请选择**报警类 (VALUE, DEV, ROC..)** 选项。
9. 要打印报警类型，请选择**报警类型 (HIHI, LO, MAJDEV,…)** 选项。

   如需有关可用报警类型的详细信息，请参阅 “报警类型”。
10. 要打印报警优先级，请选择**优先级**选项。
11. 要打印报警名 （标记名），请选择**报警名**选项。在**长度**框中，输入允许报警名使用的字符数 （最多 64 个字符）。
12. 要打印报警组名，请选择**组名**选项。在**长度**框中，输入允许报警组名使用的字符数 （最多 64 个字符）。
13. 要打印报警供应器的名称，请选择**报警供应器**选项。在**长度**框中，输入允许报警供应器名使用的字符数 （最多 64 个字符）。
14. 要打印标记名的值，请选择**报警值**选项。在**长度**框中，输入允许该值使用的字符数 （最多 32 个字符）。输入的数值应该足够大以满足精度要求。
15. 要打印标记名的报警限，请选择**极限**选项。在**长度**框中，输入允许报警限使用的字符数 （最多 32 个字符）。输入的数值应该足够大以满足精度要求。

16. 要打印与报警条件关联的操作员 ID，请选择**操作员**选项。在**长度**框中，输入允许操作员 ID 使用的字符数 （最多 16 个字符）。
17. 要打印与标记名关联的 “报警注释”，请选择**注释**选项。在**长度**框中，输入允许注释使用的字符数 （最多 131 个字符）。
18. 选择**删除尾部空格**选项，以便从打印的字段中删除在实际字段值的长度小于为该字段配置的长度时填补的额外尾部空格。

要配置报警打印功能

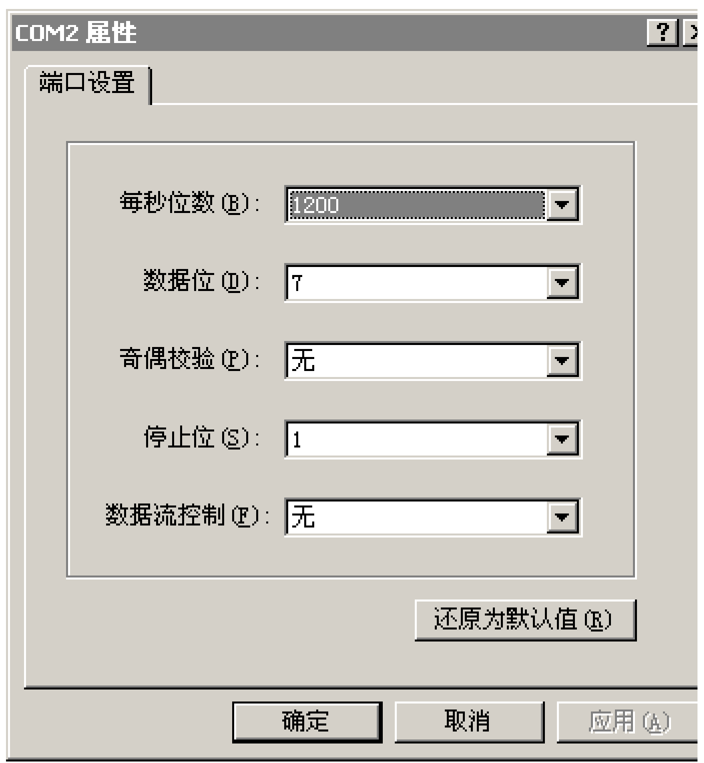
1. 单击**报警打印程序**对话框中的**配置**菜单，或单击 “报警打印程序”工具栏上的**配置报警查询**工具。
2. 此时会出现**配置设置**对话框。
3. 选择**打印**选项卡。



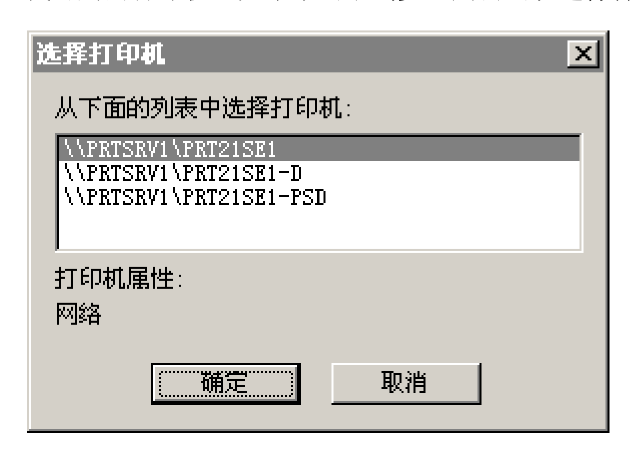
4. 为打印机选择您希望 “报警打印程序”用于打印报告的选项 - 串行端口 （COM1、 COM2 等）、并行端口 （LPT1、 LPT2 等）或 Windows 打印机。

备注 用于打印报警状态更改的打印机不得用于其它打印任务，否则会干扰 “报警打印程序”的打印功能。

5. 如果选定 COM 端口，请单击**端口配置**以配置串行端口。此时会出现 **COM# 属性**对话框。



6. 通过单击每个字段右边的箭头，并选择与当前使用的 COM 端口相适应的设置，对 COM 端口进行配置，然后单击**确定**。
7. 如果选定 Windows 打印机，请输入该打印机的名称，或单击**浏览**查找可用的打印机。如果单击了**浏览**，则会出现**选择打印机**对话框。



8. 滚动列表查找所需的打印机，单击打印机名称，然后单击**确定**。该打印机可连接到本地机器或连接到网络。任何 Windows 打印机均可用于打印 “报警打印程序”的输出。

备注 如果所需的打印机未出现在 “选择打印机”列表中，则您可以将该打印机添加到 Windows 配置中。如需有关详细信息，请参阅 Windows 操作系统文档。

运行报警打印程序的报警查询

每个查询均会记录当前打开的 “报警打印程序配置文件” (*.ALC) 中指定的所有报警；如果未指定文件，则记录配置 “报警打印程序”期间所选的当前设置指定的所有报警。

如需有关详细信息，请参阅 “配置报警打印程序”。

您可以在 “报警打印程序”上运行多项查询，每个查询均可指定不同的参数。每个查询均会由 “报警打印程序”的一个单独的实例执行。如果 “报警打印程序”的两个实例在运行相同的查询，则查询结果会完全相同。

“报警打印程序”正在运行时，您可以手工启动或停止查询，具体如下：

要启动或停止报警查询

在**查询**菜单上，选择**启动 / 停止**。

提示 要快速启动或停止查询，请单击工具栏上的 “启动 / 停止查询”工具。

要自动启动报警打印程序并打开特定文件

通过在 .BAT 文件中使用下列命令，您可以让 “报警打印程序”在系统启动时自动启动并打开特定的文件：

```
ALMPRT.EXE MYQUERY.ALC
```

其中， **MYQUERY.ALC** 是打开的 “报警打印程序”配置文件的名称。（指定 **.EXE** 扩展名属可选项）。

要自动启动报警打印程序并运行查询

为防止因系统意外关机或重新启动而遭致丢失任何查询数据，您可以通过在 .BAT 文件中使用下列命令，让 “报警打印程序”自动启动并运行特定的查询：

```
ALMPRT.EXE -q MYQUERY.ALC
```

通过在命令中使用 -q，查询便会在系统启动时自动运行。 （指定 **.EXE** 扩展名属可选项）。

Alarm DB Logger 实用程序

“InTouch 分布式报警”系统包含 Alarm DB Logger 实用程序，可将报警与事件记录到 Microsoft SQL Server 或 Microsoft Data Engine (MSDE) 数据库。

MSDE 是 SQL Server 的简装版，但它有自己的独到之处。尤其是其易用性，您可以将最初为 MSDE 建立的数据库轻松连接到完整的 SQL Server 服务中。这样便没有必要升级数据库或是将数据库中各个单独的表从 MSDE 复制到完整版的 SQL Server 之中。对于部署大量计算机资源并不经济适用的环境，这便使得它非常适合。 MSDE 数据库的最大尺寸为 2 GB。

Alarm DB Logger 是一个 “报警接收器”。您可以使用定义要记录的报警的**报警查询**来配置它。您可以使用 Alarm DB Logger 指定报警查询并记录产生的报警记录。这些报警查询会通过 “分布式报警系统”的 “报警接收器”接口发送。

Alarm DB Logger 还具有自动重新连接的功能。丢失与数据库的连接时，Logger 会按固定的时间间隔继续检查数据库连接。在重新建立连接之后，记录工作将继续进行。

无论是作为服务还是普通应用程序运行，Alarm DB Logger 均会报告所有的错误。

Alarm DB Logger 由下面两个组件组成：

1. **Alarm DB Logger Manager 实用程序** - 此实用程序是一个单独的可执行文件，它只负责启动与停止记录操作。在它启动，它既可以作为服务又可以作为普通的应用程序开始工作 （具体取决于您在 “Alarm DB Logger Manager”中选择的运行模式）。记录实用程序从注册表检索设置信息并执行记录操作。
2. **Alarm DB Logger 配置实用程序** - 此实用程序负责用户输入与数据库配置。“Alarm DB Logger Manager”可供选择 Alarm DB Logger 的运行方式 （既可以作为 Windows 服务，也可以作为普通的应用程序）。

备注 “Alarm DB Logger Manager”只将设置值保存到注册表。此实用程序负责启动与停止 Alarm DB Logger。它还负责显示 “智能缓存”的状态。如果在 wwalmlogger.exe 运行时关闭 “Alarm DB Logger Manager” (almlogwiz.exe) （按 Esc 键或单击对话框右上角的 "X" 按钮），记录操作不会停止。

进度条状态会指出加载到内存缓冲区的报警记录百分比。在 SQL Server 连接中断和 （或）报警的发生速率高于 Alarm DB Logger 的记录速率时，会将这些报警缓存起来。

“Alarm DB Logger 配置实用程序”可供您：

- 将应用程序作为 Windows 服务或普通的应用程序来运行
- 选择数据库连接类型 - SQL Server 或 MSDE
- 在数据库中创建必要的 SQL 表
- 指定将成为记录实例一部分的报警查询
- 选择记录模式 - 详细或合并
- 启用 / 禁用记录事件的功能

- 设置性能调整能参数 - 自动重新连接速率与性能调整参数不同。它取决于与 SQL Server 尝试建立连接的超时值。
- 将设置存储到注册表

如需有关详细信息，请参阅 “Alarm DB Logger 配置”。

记录到 SQL 数据库

Alarm DB Logger 会将报警数据记录到数据库。如果 OLEDB 供应器是 SQL Server，则您需要在 “Alarm DB Logger Manager”中指定 SQL Server 机器。如果数据库中不存在所需的数据结构，则 Alarm DB Logger 会自动创建它们。

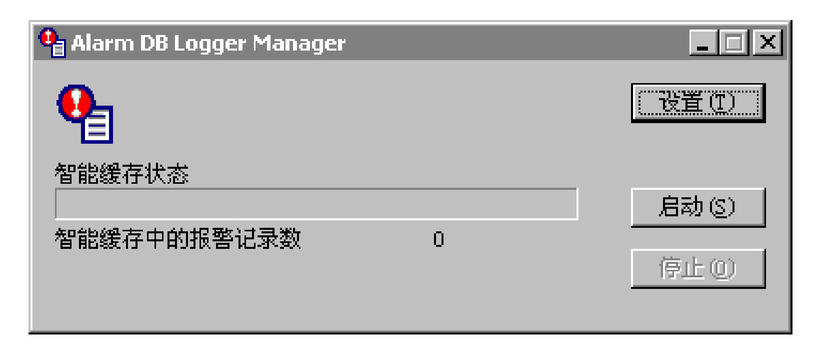
如需有关详细信息，请参阅 “分布式报警数据库视图”。

使用 Alarm DB Logger 实用程序

Alarm DB Logger 实用程序是一个单独的可执行文件，它负责记录工作。它启动之后，既可作为服务又可作为普通的应用程序开始工作 （具体取决于在配置 Alarm DB Logger 时选择的运行模式）。记录实用程序从注册表检索设置信息并执行记录操作。 Alarm DB Logger 是一个 “报警接收器”。

要使用 Alarm DB Logger 实用程序

1. 启动 “Alarm DB Logger Manager”。此时会出现 **Alarm DB Logger Manager** 对话框。



提示 最小化时，“Alarm DB Logger Manager”显示为系统工具栏中的一个图标。右击该图标时，会出现一个弹出式菜单，显示下列命令：

启动 - 开始报警记录过程。
停止 - 结束报警记录过程。
设置 - 打开 **Alarm DB Logger Manager - 配置**对话框。
隐藏窗口 - 将 “Alarm DB Logger Manager”对话框最小化，使之成为系统工具栏中的一个图标。
显示窗口 - 打开并最大化 “Alarm DB Logger Manager”对话框。
关闭 - 退出 “Alarm DB Logger Manager”实用程序。

智能缓存状态会指出加载到内存缓冲区的报警记录百分比。

2. 单击**设置**来配置 Alarm DB Logger。此时会出现 **Alarm DB Logger Manager - 配置**对话框。

   如需有关配置 Alarm DB Logger 的详细信息，请参阅 “Alarm DB Logger 配置”。

3. 单击**启动**开始报警记录过程。
4. 单击**停止**结束报警记录过程。

Alarm DB Logger 配置

在开始使用 Alarm DB Logger 之前，必须配置若干项目，如数据库连接、查询列表及记录模式等等。此信息可通过 “Alarm DB Logger Manager”进行配置。

第一个“Alarm DB Logger 过滤器”实用程序对话框会处理数据库连接。您可以选择 SQL Server 或 Microsoft Data Engine (MSDE) 来存储数据。“Alarm DB Logger 配置实用程序”也可供创建所需的数据结构 （表、视图及存储过程）及测试数据库连接。

使用 “Alarm DB Logger Manager”创建数据库、表及存储过程时，同时会创建三个具有不同身份验证 / 安全级别的用户帐号。该 “管理器”与 IndustrialSQL Server 具有相同的身份验证 / 安全级别。这三个级别是：

帐户	口令
wwAdmin	wwadmin
wwPower	wwpower
wwUser	wwuser

要配置 Alarm DB Logger

1. 启动 “Alarm DB Logger 过滤器”。此时会出现 **Alarm DB Logger Manager** 对话框。

2. 单击**设置**。此时会出现 **Alarm DB Logger Manager - 配置**对话框。



3. 在 **SQL Server/MSDE** 组中，单击**服务器名**箭头，以打开可用 SQL/MSDE 服务器的列表，然后选择要使用的数据库服务器的名称。

4. 在**用户信息**组的相应输入框中，输入您的**用户名**与**口令**。（基于安全方面的考虑，所有**口令**字符均显示为 **）。

5. 在**记录模式**组中，单击**详细**或**合并**单选钮。

   合并模式可降低数据库大小。将报警记录到数据库时，合并模式下的报警与事件记录仍继续显示原始数据，因此每次更改并没有生成新的记录。 在 “详细”模式下，记录会显示事件发生时的数据。

   在 “合并模式”下，第一个 UNACK_ALM 表示发生报警，第二个 UNACK_ALM 代表最近的子状态改变（如果有）。 如果报警只经历一次子状态改变，则只会看到一条 UNACK_ALM 记录。否则，您会看到有两条 UNACK_ALM 记录。 不过，如果报警经过数次子状态改变 （如 Hi->HiHi->Hi），则第二条 UNACK_ALM 记录将只代表最终的子状态。

   选定**返回意味着确认**并且报警未经确认便返回正常状态时，ACK_RTN 与 ACK 记录会同时在 “合并”与 “详细”记录模式下创建。

重要！ 如果您希望切换**详细**与**合并**模式，则必须重新创建数据库，并且会丢失现有的全部数据。创建数据库时会覆盖全部现有的报警数据库信息。

---

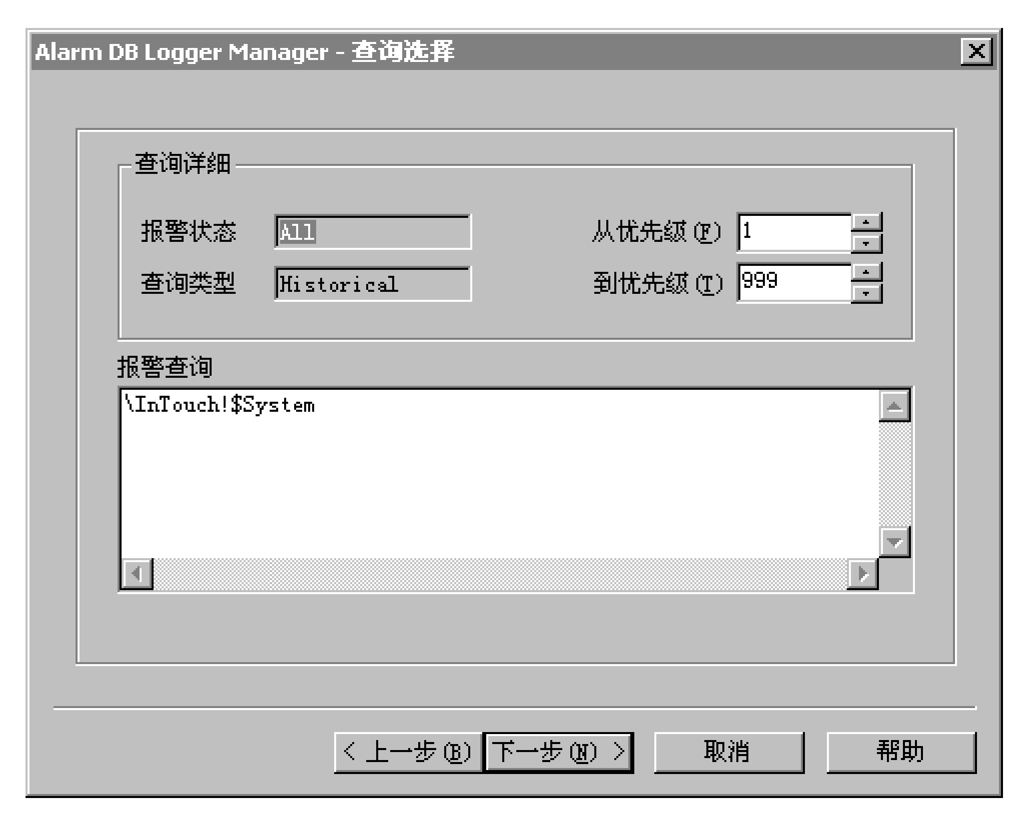
6. 单击**测试连接**来测试与目标数据库的连接。
7. 单击**创建**在选定的“服务器”上创建新的数据库 / 表。
8. 单击**下一步**访问 **Alarm DB Logger Manager - 查询选择**对话框。

Alarm DB Logger 查询配置

第二个“Alarm DB Logger 过滤器”对话框显示要为报警记录选取的查询详细资料。它提供一个多行文本输入框供您输入查询。它还可供您选择**从优先级**与**到优先级**值。

要配置查询详细资料

1. 如果“Alarm DB Logger Manager 实用程序”尚未运行，请启动它。此时会出现 **Alarm DB Logger Manager 配置**对话框。
2. 单击**下一步**。此时会出现 **Alarm DB Logger Manager - 查询选择**对话框。



备注 **报警状态**字段显示要记录的报警状态。**查询类型**字段显示查询的类型。（这两者都是只读字段）。

3. 单击**从优先级**箭头以选择报警优先级范围的起始值。
4. 单击**到优先级**箭头以选择报警优先级范围的结束值。
5. 在多行**报警查询**输入框中，输入要执行的 InTouch 报警查询组。

6. 单击下一步。此时会出现 **Alarm DB Logger Manager - 高级设置**对话框。

Alarm DB Logger 高级设置配置

第三个 “Alarm DB Logger Manager”对话框显示高级查询设置，如运行模式（作为 Windows 服务作为还是普通的应用程序）。它还提供用于记录事件的选项。您可以通过设置从内存缓存区到数据库的报警刷新频率，来调整报警记录功能的性能。

要配置高级设置详细资料

1. 如果 “Alarm DB Logger Manager”尚未运行，请启动它。此时会出现 **Alarm DB Logger Manager 配置**对话框。
2. 单击下一步。此时会出现 **Alarm DB Logger Manager - 查询选择**对话框。
3. 单击下一步。此时会出现 **Alarm DB Logger Manager - 高级设置**对话框。



4. 在**记录器运行方式为**组中，选择要用于记录器的选项。
   - **Windows 服务** - 将记录器配置成作为 Windows 服务运行。
   - **正常应用程序** - 将记录器配置成作为普通的应用程序运行。
5. 选择**记录事件**选项以便将所有 “事件”记录到表中。
6. 在**每隔 # 毫秒记录报警**输入框中，输入以毫秒计的报警记录间隔。

备注 重新连接速率与此毫秒速率设置不同。毫秒速率是读取报警的间隔时间。重新连接速度取决于与 SQL Server 尝试建立连接的超时值。

将 “性能调整”参数设置得太低会影响系统性能。

7. 单击**完成**关闭 “Alarm DB Logger Manager”实用程序，并将所有的记录器配置保存到注册表。

Alarm DB Logger 的新增功能

InTouch 有三种安全模式。这些安全模式分别是 “基于操作系统的安全性”、“InTouch 安全性”及 “ArchestrA 安全性”。将安全模式设置为 Windows 安全性时，报警供应器会将操作员的全名和域名作为报警记录的一部分而发送。随后，Alarm DB Logger 在报警数据库中存储这些附加字段。

InTouch Alarm DB Logger 也会将 “未确认持续时间”值存储到报警数据库。 “未确认持续时间”是 ACK 时间减去最近报警状态改变时间，其中所有的日期与时间均用 GMT 表示。只要某个报警记录被确认，Alarm DB Logger 便会按如下方式计算 “未确认持续时间”：

如果报警经历一次子状态改变 （如从 Hi 变为 HiHi），则 “未确认持续时间”是 ACK 时间减去子状态时间。否则，“未确认持续时间”会是 ACK 时间减去报警发生时间。

如果报警记录状态改变类型是 ACK 之外的其它任何类型，则 Alarm DB Logger 会记录 “未确认持续时间”列的缺省值 （空值）。

Alarm DB View 控件也会显示返回持续时间。这是用返回时间减去报警发生时间计算得出的。报警持续时间不会存储到数据库，但可以根据需要计算出来，并由数据库视图显示。

如果 Alarm DB Logger 尚未运行，则 InTouch 及其相关应用程序 “也不会”去启动它。

报警数据库清除 / 归档实用程序

“报警数据库清除 / 归档”实用程序是随 InTouch 一起安装的独立应用程序。 “报警数据库清除 / 归档”实用程序负责配置报警数据库的 “清除 / 归档”操作，并显示任何清除活动的状态。此实用程序还可以提供基于特定计时器的清除与归档功能。

您可以使用 “报警数据库清除 / 归档”实用程序来：

- 选择需要清除和 （或）归档的记录类型 （合并报警记录或报警详细记录）。
- 根据配置指定自动定时清除 / 归档作业。
- 在必要时清除整个数据库。
- 设置执行清除 / 归档活动的时间。

- 指定执行清除 / 归档活动的时间间隔。这可以是每天、每周或每月。
- 显示清除 / 归档作业的状态。
- 将状态信息保存到日志文件。（将来您可以使用此日志文件来检查清除 / 归档活动的状态）。

清除 / 归档通用属性配置

“报警数据库清除 / 归档”实用程序的**通用**属性页用于设置 “清除 / 归档”属性。您可以选择要清除的表类型 - **AlarmDetail** 或 **AlarmConsolidated** 表。同时您还有归档要清除的数据的选项。您可以选择归档数据的文件夹路径。对于清除 / 归档的每个表，都会创建归档文件。归档文件名是自动创建的，使用的名称是表名、日期及时间的唯一组合。例如，在 2000 年 11 月 10 日下午 5:30 归档的 **AlarmMaster** 的归档文件的名称是 **AlarmMaster_11102000_1730.txt**。

如需有关详细信息，请参阅 “分布式报警数据库视图”。

要清除 / 归档报警数据库

1. 启动“报警数据库清除 / 归档”实用程序。此时会出现**清除 / 归档**对话框。



最小化时，“报警数据库清除 / 归档”实用程序显示为系统工具栏中的一个图标。右击该图标时，会出现一个弹出式菜单，显示下列命令：-

命令	描述
立即全清	清除整个数据库。
立即清除	使用现有的设置立即开始清除。
取消清除	停止当前的清除过程。（此命令仅当清除作业正在进行时才会处于活动状态）。停止清除时，数据库将恢复为最初的状态。
启动	激活基于计时器的清除作业。
停止	停止基于计时器的清除作业。
立即测试	执行清除测试来验证与数据库及目标位置的连接
清除状态	清除状态窗口。
隐藏窗口	最小化“报警数据库清除 / 归档”实用程序，使之成为系统工具栏中的一个图标。

命令	描述
显示窗口	打开并最大化 “报警数据库清除 / 归档”实用程序。
退出	关闭 “报警数据库清除 / 归档”实用程序。

如果右击 “报警数据库清除 / 归档”配置属性页上的任意位置，则会出现相同的弹出式菜单。

2. 在**清除属性**组中，选择要用于清除的选项：
   - **详细模式**可清除在 “详细模式”下记录的报警。
   - **合并模式**可清除在 “合并模式”下记录的报警。
3. 在选择**清除属性**模式之后，相应的**在线天数**输入框会变为活动状态。输入您希望保留数据的天数。在指定天数之前的所有数据都会被清除 （有效值是 0 - 9999）。如果该数字为 0，则除当前记录之外的其它所有记录均会被清除。
4. 选择**归档**选项，然后通过在**归档文件夹路径**输入框中输入完整路径，或通过单击相应的 “浏览” (...) 按钮查找并选择文件，来指定您要归档到其中的数据库文件的完整路径 （最长可达 225 个字母数字字符）。（选定之后，该路径将自动显示在输入框中）。
5. 在**日志文件路径**输入框中，指定用于存储状态文件的文件夹的完整路径（最长可达 225 个字母数字字符），或单击相应的 “浏览” (…) 按钮来查找并选择文件夹。（选定之后，该路径将自动显示在输入框中）。

备注 缺省条件下，日志文件 (wwalmpurge.log) 存储在 FactorySuite\Common 文件夹中，但可以由用户改写。每次执行清除 / 归档活动时，会将记录追加到此文件中。

6. 单击**应用**将设置保存到注册表，或单击**确定**保存设置并关闭 “报警数据库清除 / 归档”实用程序。

清除 / 归档数据库配置

“报警数据库清除 / 归档” **数据库**属性页可供您选择清除 / 归档功能将要访问的服务器。

要配置报警数据库清除 / 归档实用程序数据库

1. 启动 “报警数据库清除 / 归档”实用程序。此时会出现**清除 / 归档**对话框。

2. 单击**数据库**选项卡来激活**数据库**属性。

如果右击属性页，则会出现一个弹出式菜单，显示下列命令：

命令	描述
立即全清	清除整个数据库。
立即清除	使用现有的设置立即开始清除。
取消清除	停止当前的清除过程。（此命令仅当清除作业正在进行时才会处于活动状态）。停止清除时，数据库将恢复为最初的状态。
启动	激活基于计时器的清除作业。
停止	停止基于计时器的清除作业。
立即测试	执行清除测试来验证与数据库及目标位置的连接
清除状态	清除状态窗口。
隐藏窗口	最小化 “报警数据库清除 / 归档” 实用程序，使之成为系统工具栏中的一个图标。
显示窗口	打开并最大化 “报警数据库清除 / 归档” 实用程序。
退出	关闭 “报警数据库清除 / 归档” 实用程序。

3. 在 **SQL Server/MSDE** 组中，单击**服务器名**箭头，以打开可用 SQL/MSDE 服务器的列表，然后选择要使用的数据库服务器的名称。
4. 此时**数据库**框会显示该数据库名。（这是一个只读字段，缺省值为 **WWALMDB**）。
5. 在**用户信息**组的相应输入框中，输入您的**用户名**与**口令**。（基于安全方面的考虑，所有**口令**字符均显示为 **）。
6. 单击**测试连接**来测试与数据库的连接。
7. 单击**应用**将设置保存到注册表，或单击**确定**保存设置并关闭 “报警数据库清除 / 归档” 实用程序。

清除 / 归档配置

通过使用 “报警数据库清除 / 归档” 实用程序的**清除 / 归档**属性页，您可以设置触发自动清除与归档活动的时间。它还可供您执行清除测试来验证与数据库及目标位置的连接，并可启动与停止清除作业。

要配置报警数据库清除 / 归档实用程序时间间隔

1. 启动 “报警数据库清除 / 归档” 实用程序。此时会出现**清除 / 归档**对话框。

2. 单击**清除 / 归档**选项卡来激活**清除 / 归档**属性页。

如果右击属性页，则会出现一个弹出式菜单，显示下列命令：

命令	描述
立即全清	清除整个数据库。
立即清除	使用现有的设置立即开始清除。
取消清除	停止当前的清除过程。（此命令仅当清除作业正在进行时才会处于活动状态）。停止清除时，数据库将恢复为最初的状态。
启动	激活基于计时器的清除作业。
停止	停止基于计时器的清除作业。
立即测试	执行清除测试来验证与数据库及目标位置的连接
清除状态	清除状态窗口。
隐藏窗口	最小化 "报警数据库清除 / 归档" 实用程序，使之成为系统工具栏中的一个图标。
显示窗口	打开并最大化 "报警数据库清除 / 归档" 实用程序。
退出	关闭 "报警数据库清除 / 归档" 实用程序。

3. 选择要使用的**时间间隔**选项：

- **每天** - 每天从**时间**字段中选定的时间开始执行清除作业。
- **每星期** - 每星期从**日期**字段中选定的日期及 “时间” 字段中选定的时间开始执行清除作业。
- **每月** - 每月从**日期**字段中选定的日期及**时间**字段中选定的时间开始执行清除作业。

备注 日期的缺省值为空白。如果您选择**每星期**或**每月**，则必须选择日期。

4. 在选定**时间间隔**选项后，单击**时间**向上 / 向下箭头来选择当天开始清除 / 归档活动的时间。
5. 在选定**时间间隔**选项 （仅限于**每星期**或**每月**）之后，单击**日期**箭头来选择执行清除 / 归档活动的日期。 （在**时间间隔**为**每星期**时指星期几，在**时间间隔**为**每月**时指日期）。
6. 清除活动的状态显示在**状态**区。
7. 单击**立即清除**可使用现有的设置立即执行清除。

备注 单击**立即清除**会改写激活时间，并立即执行清除与归档作业。**立即清除**会检查是否存在归档文件并将内容追加到相同的文件中。如果不存在归档文件，则会按照命名惯例创建一个文件，然后将此文件用于归档。**立即清除**不会删除表中的项目，如 **ProviderSession**、 **Query** 以及 **Cause** 等，这些项目通过外部关键字约束条件与 **AlarmMaster** 等主表链接。这些表中相关的记录会写入文件，以保持数据的一致性，同时也会保存在数据库中。

8. 单击**立即测试**以执行清除测试，检查与数据库及目标位置的连接。（仅当在 “通用” 选项卡上选定**归档**选项时，此按钮才会变为活动状态。它会在指定的归档路径中创建一个空白归档文件）。
9. 单击**取消清除**以停止清除作业。（仅当清除正在进行时，此按钮才会处于活动状态）。停止清除时，数据库将恢复为最初的状态。
10. 单击**立即全清**以清除整个数据库。

注意！ 仅当 Alarm DB Logger 不在运行时，才可选择**立即全清**。在 Alarm DB Logger 正在向数据库记入数据时，如果成功提交**立即全清**，则 Alarm DB Logger 会停止将数据记入数据库，并开始缓存记录。

11. 单击**应用**将设置保存到注册表，或单击**确定**保存设置并关闭 “报警数据库清除 / 归档” 实用程序。

报警数据库恢复实用程序

“报警数据库恢复” 实用程序用于将备份文件中归档的报警记录恢复到数据库中。

“报警数据库恢复” 实用程序可协助您将归档的数据恢复到数据库中，以便执行进一步的操作。

恢复数据库配置

配置属性页用于指定要将数据恢复到其中的数据库。如果指定的数据库不在服务器上，则会提示您在服务器上创建一个新的数据库。确认之后，会在指定的服务器上使用缺省参数创建一个新数据库。

要配置数据库以便恢复：

1. 启动**报警数据库恢复**实用程序。此时会出现**恢复**对话框，并显示**配置**选项卡。



最小化时，“报警数据库恢复”实用程序显示为系统工具栏中的一个图标。右击该图标时，会出现一个弹出式菜单，显示下列命令：

命令	描述
恢复	开始恢复过程。
取消恢复	结束恢复过程。
清除状态	清除状态窗口。
隐藏窗口	最小化 “报警数据库恢复”实用程序，使之成为系统工具栏中的一个图标。
显示窗口	打开并最大化 “报警数据库恢复”实用程序。
退出	关闭 “报警数据库恢复”实用程序。

如果右击 “报警数据库恢复”配置属性页上的任意位置，则会出现相同的弹出式菜单。

2. 在 **SQL Server / MSDE** 组中，单击**服务器名**箭头，以打开可用 SQL/MSDE 服务器的列表，然后选择要使用的数据库服务器的名称。
3. 此时**数据库名**框会显示该数据库的名称。（这是一个只读字段，缺省值为 WWALMDB）。
4. 在**用户信息**组的相应输入框中，输入您的**用户名**与**口令**。（基于安全方面的考虑，所有**口令**字符均显示为 **）。
5. 单击**测试连接**来测试与数据库的连接。

恢复文件名配置

“报警数据库恢复”实用程序的**选择**属性页用于指定要对其执行恢复作业的文件名。它还显示一个状态列表框，显示将数据恢复到数据库的进度状态。从这个对话框中，您可以取消正在进行中的恢复操作。如果取消恢复，则数据库将恢复成最初的状态。

要选择数据库以便恢复：

1. 启动“报警数据库恢复”实用程序。此时会出现**恢复**对话框，并显示**配置**选项卡。

2. 单击**选择**选项卡以激活**选择**属性页。



如果右击属性页，则会出现一个弹出式菜单，显示下列命令：

命令	描述
恢复	开始恢复过程。
取消恢复	结束恢复过程。
清除状态	清除状态窗口。
隐藏窗口	最小化“报警数据库恢复”实用程序，使之成为系统工具栏中的一个图标。
显示窗口	打开并最大化“报警数据库恢复”实用程序。
退出	关闭“报警数据库恢复”实用程序。

3. 在**归档文件目录**输入框中，输入归档文件位置的完整路径（最长可达 255 个字母数字字符），或单击相应的“浏览”(...) 按钮来查找并选择要创建及存储归档文件的文件夹。（选定之后，该路径将自动显示在输入框中）。

4. 在**日志文件目录**输入框中，输入其中将创建与存储日志文件的完整路径（最长可达 225 个字母数字字符），或单击相应的“浏览”(...) 按钮来查找并选择文件夹。（选定之后，日志文件的完整路径将自动出现在输入框中）。

5. 如果要在目标数据库中重新创建表，请选择**重建表**。如果不选择此选项，则数据会被追加到现有的表中。

注意！ 重建表会导致丢失现有的全部数据。

6. 选择**重建表**选项时，**记录模式**组会变为活动状态。选择要使用的模式：
   - **详细** - 仅恢复详细表
   - **合并** - 仅恢复合并表
7. 在**恢复此后的文件 (日期 / 时间)** 字段中，选择在此后执行恢复的日期与时间。

备注 日期指定在此后执行恢复的日期。时间指定在指定日期开始恢复的时间。这两个字段的缺省值为当前日期与时间。

AlarmSuite History Migration 实用程序

AlarmSuite History Migration 实用程序随 InTouch 一起安装。AlarmSuite History Migration 实用程序负责将 AlarmSuite 数据从 AlarmSuite 数据库迁移到新的 “InTouch 分布式报警系统”的数据库架构。

如需有关详细信息，请参阅 “分布式报警数据库视图”。

要迁移 AlarmSuite 历史数据：

1. 启动 AlarmSuite History Migration 实用程序。此时会出现 **AlarmSuite History Migration 实用程序**对话框。



2. 在 **AlarmSuite 源数据库**组中，单击 **DSN** 箭头，以打开本地机器中配置的 ODBC DSN 名称列表。
3. 选择要使用的 ODBC DSN 名称。
4. 在**用户名**输入框中，输入您的 DSN 用户名。
5. 在**口令**输入框中，输入您的 DSN 用户口令。（基于安全方面的考虑，所有输入均显示为 **）。
6. 选择要使用的**清除源数据库**选项：
   - **是** - 在数据迁移结束时清除源数据库。（创建表是事务处理的一部分，但创建数据库却不是。如果您选择**是**，则会在从源数据库迁移数据之后提交清除事务）。
   - **否** - 在数据迁移结束时不清除源数据库
7. 在**目标数据库**组中，单击**服务器名**箭头 （在 **SQL Server / MSDE** 组中），以打开可用 SQL/MSDE Server 的列表。选择要使用的数据库服务器的名称。
8. 此时**数据库名**框会显示该数据库的名称。（这是一个只读字段，缺省值为 WWALMDB）。
9. 在**用户名**输入框中，输入您的 SQL Server 用户名。
10. 在**口令**输入框中，输入您的 SQL Server 口令。 （基于安全方面的考虑，所有**口令**字符均显示为 **）。
11. 如果要在目标数据库中重新创建表，请选择**重建表**。如果不选择此选项，则数据会被追加到现有的表。

    备注 重建表会导致丢失现有的全部数据。

12. 选择**重建表**选项时，**记录模式**组会变为活动状态。选择要使用的模式：
    - **详细** - 仅恢复详细表
    - **合并** - 仅恢复合并表
13. 单击**测试连接**来测试同来源及目标数据库的连接。
14. 单击**开始**，以便开始将数据从源数据库迁移到目标数据库。
15. 单击**停止**，以便中止进行中的迁移过程。这将回滚事务处理。 （单击**关闭**相当于单击**停止**）。

    备注 创建表是事务处理的一部分，但创建数据库却不是。如果您已将**清除源数据库**选项设置为**是**，在从源数据库迁移数据之后将提交清除事务。

16. 迁移过程完成之后，请单击**关闭**以退出实用程序。

    备注 数据迁移正在进行时，不得使用来源或目标数据库来记录新数据。这可能会导致失去数据完整性，从而导致搜索结果错误。

第 12 章

实时趋势和历史趋势

InTouch 提供了两种趋势显示对象："实时趋势" 和 "历史趋势"。您可以对两种趋势对象进行配置，以显示特定时间内多个标记名的图形表示。"实时" 趋势可供您最多用四支笔 （数据值）画图，而 "历史" 趋势则可供您最多用八支笔画图。这两种趋势都是使用 WindowMaker 中的特殊工具来创建的。 InTouch 还让您可以完全控制趋势的配置。例如，您可以指定时间间隔、值范围、网格分辨率、时间标签位置、笔数，以及颜色属性。

FactorySuite Productivity Pack 包括一个可供您用 16 支笔画图的 "笔配置网格"。如需有关详细信息，请参阅 *Productivity Pack 用户指南*。

InTouch 还支持分布式历史系统，可供您从任何 InTouch 历史记录文件中检索历史数据，甚至跨网络进行检索。

除绘制趋势图的功能之外， InTouch 还包括一个 HistData 实用程序，它设计用于处理 InTouch 历史记录文件。 HistData 实用程序可以将加密的历史记录文件 (.lgh) 转换成逗号分隔变量 (.csv) 文件，以用于在 Microsoft Excel 之类的电子表格或文本编辑环境中。

目录

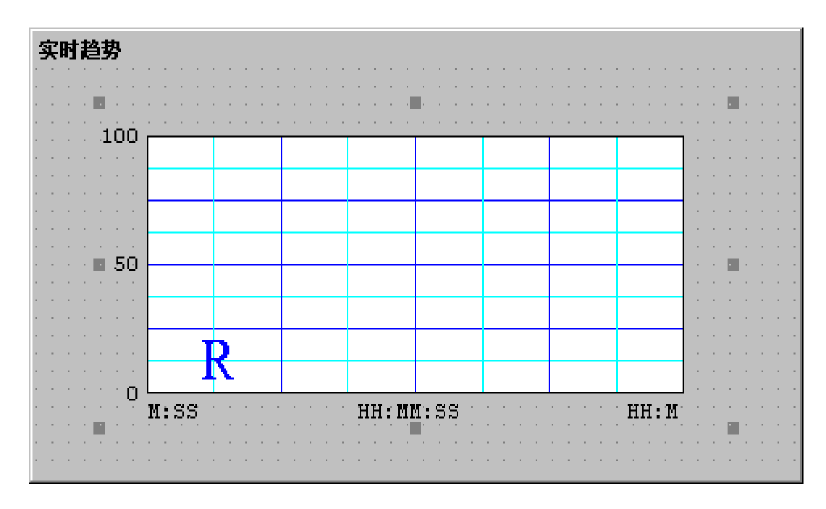
实时趋势

实时趋势是动态的。它们在运行期间不断更新。它们最多可以绘制四个本地标记名或表达式的变化。

创建实时趋势

要创建实时趋势

1. 选择**绘图对象工具栏**中的实时趋势工具。
2. 在窗口中单击鼠标左键，然后对角拖动鼠标，以绘制一个所需趋势大小的矩形框。（您可以绘制任意大小的趋势图，而且可以在屏幕上放置的趋势图数目不受限制）。
3. 释放鼠标。此时在窗口中会出现实时趋势对象：



   在运行期间，数据会自右向左写入趋势图。

4. 双击趋势以打开它的配置对话框。

   趋势对象类似于在 WindowMaker 中绘制的任何其它对象。您可以用鼠标拖曳来移动它，也可以通过抓住对象的一个 “手柄”来调整它的大小。在窗口中，您可以放置多个趋势图。

配置实时趋势

第一次粘贴实时趋势对象时，系统将使用缺省配置。在实时趋势配置完毕之后，您创建的下一个趋势将使用相同的设置作为缺省配置。

要配置实时趋势

1. 双击趋势，或选定趋势，然后在**特别**菜单上，单击**动画链接**。此时会出现**实时趋势配置**对话框。



提示 如果在实时趋势配置对话框中右击文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

备注 “实时趋势配置”对话框中所有的项目均与趋势大小无关，并且不得在运行时修改。

2. 在**时间间隔**框中，输入要在趋势上显示的水平 （x 轴）时间长度，然后选择时间长度的时间增量选项。

   例如，如果在**时间间隔**框中输入 30，然后选择**分钟**，则图表的水平时间范围会是 30 分钟。

3. 在**样本间隔**框中，输入求算趋势表达式并更新图表的频率，然后选择与此数值相关的时间增量选项。

   例如，如果在**间隔**框中输入 10，并选择**秒**作为时间增量，则表达式将每隔 10 秒求值一次。

4. 在**颜色**组中，单击**图表颜色**框以打开 InTouch 调色板。单击调色板中要用作趋势背景的颜色。
5. 在**颜色**组中，单击**边框颜色**框以打开 InTouch 调色板。单击调色板中要用作趋势边框的颜色。

提示 对所有的颜色选项重复此过程。

6. 在**时间刻度数**组的**主刻度数**框中，输入趋势所需的主时间刻度数，然后选择要用于刻度线的颜色。

   主时间刻度数之间的最大时间是 65536 秒或 18 小时 12 分钟 16 秒。

   主时间刻度数的数值必须是**副 / 主刻度数**的偶数倍。

7. 在**时间刻度数**组的**副 / 主刻度**框中，输入每个主时间刻度内可见的副时间刻度数，然后选择要用于刻度线的颜色。
8. 如果要在趋势的顶部显示时间标签，请在**时间刻度数**组中，选择**顶部标签**。
9. 如果要在趋势的底部显示时间标签，请在**时间刻度数**组中，选择**底部标签**。

   趋势既可以同时包含顶部标签和底部标签，也可以没有任何标签。

10. 如果使用了时间标签，请在**时间刻度数**组的**副 / 主刻度**框中，输入趋势的每个主时间刻度线的时间标签数。
11. 在**时间刻度数**组中，选择要用于主时间刻度线的颜色。
12. **值刻度数**组的中的设置与**时间刻度数**组中的设置采用相同方式进行配置，不同的是副刻度和主刻度会设置趋势的垂直值 （y 轴）范围。此范围使用 “工程单位”，并且对于所有要绘制趋势的标记名都是相同的。

提示 要在运行时显示副值和主值刻度的小数点，必须在此处设置它们的格式。例如， 0.00 - 100.00。

13. 在**表达式**框中，输入要用每支**笔**绘制其趋势的本地标记名或表达式。

    趋势图中最多可见到 4 支笔。这些笔可用于显示任何本地标记名或包含一个或多个本地标记名的表达式。（无法记录或绘制消息型标记的趋势）。趋势表达式对于创建自定义的显示画面很有用处，它可用于显示各种范围的标记名。

14. 单击颜色框，选择用于在趋势中绘制每个标记名的每支笔的颜色。

15. 在**宽度**框中，输入每支笔的象素宽度值。选择大于 1 的笔宽会显著降低更新和打印趋势时的性能。
16. 单击**选择显示字体**以访问**字体**对话框，选择要在打印趋势时使用的字体、样式和大小。
17. 如果要使趋势仅当它显示在活动窗口中时才更新，选择**只在载入内存时更新**。

    如果不选择此选项，则即使不在打开的窗口中，趋势也总是会去更新。如此会导致整个系统的性能有稍微降低。
18. 单击**确定**。

要增强实时趋势性能

1. 将笔宽设置为 '1'。
2. 确保 “实时” 趋势上面没有放置其它对象。
3. 减少要采集的 “样本” 数。

例如，如果将 “时间间隔” 设置为 30 分钟，且将 “采样间隔” 设置为 2 秒，则 30 分钟内采集的样本数会按如下公式计算：

```
30* 60/2 = 900
```

如果将 “时间间隔” 设置为 30 分钟，且将 “采样间隔” 设置为 5 秒，则 30 分钟内采集的样本数会按如下公式计算：

```
30* 60/5 = 360
```

历史趋势

历史趋势可以为您提供一个历史时间和日期的数据 “快照”。它们不是动态的。与实时趋势不同，历史趋势只在接到指示时才会更新，例如通过执行 QuickScript 或由操作员执行操作 （如单击按钮）时。

一次最多可以给八个标记名 （笔）绘制趋势，但对显示的趋势数目则没有任何限制。您可以充分灵活地设计趋势界面。您可以创建一个 “指示器”，供操作员在趋势图上 “移动”，以根据指示器的当前位置来访问各种数据。例如，在操作员将指示器放在有可见数据的趋势区时，该位置处被绘制趋势的所有数据库值的时间和值均会被返回给您。

您也可以创建按钮在指示器或数据 （例如最大值到最小值）之间进行放大或缩小。您可以显示整个图表或指示器之间区域的平均值和标准偏差。历史趋势也可以滚动任意时间量。您可以创建自定义刻度，并将它们链接到 **.MinEU** 和 **.MaxEU** “标记名点域”，以显示最小和最大 “工程单位”。

分布式历史系统将历史趋势的检索能力扩展到了远程记录数据库。此系统允许在单个趋势中显示多个历史记录数据库的信息。

备注 您必须为 “标记名字典”中的每个标记名选择 “记录数据”选项，以便绘制其趋势。

如需有关记录标记名的详细信息，请参阅 “记录标记名”。

另请参阅 “配置历史记录属性”。

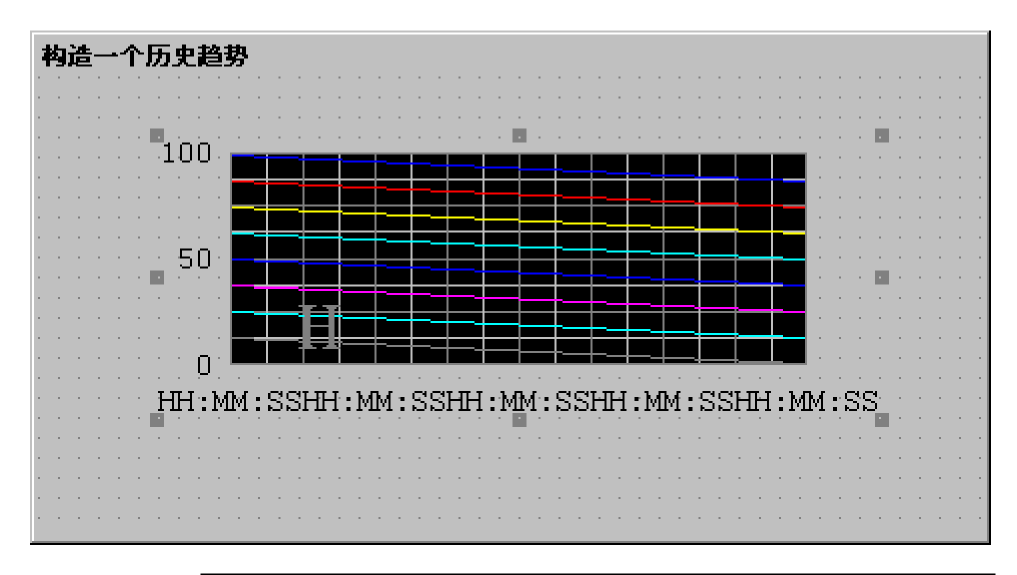
创建历史趋势

要创建历史趋势

1. 选择**绘图对象工具栏**中的历史趋势工具。历史趋势工具用于绘制历史趋势对象。
2. 在窗口中单击，然后对角拖动鼠标以绘制一个所需趋势大小的矩形框。

   提示 您可以绘制任意大小的趋势图。您也可以在窗口中放置多个趋势。

3. 释放鼠标。此时历史趋势会出现在窗口中：



提示 趋势对象类似于在 WindowMaker 中绘制的任何其它对象。您可以用鼠标抓住趋势对象来移动它，也可以通过抓住它的任何 “手柄”来调节其大小。

配置历史趋势

第一次粘贴历史趋势对象时，将会使用系统的缺省配置。在趋势配置完毕之后，下次创建的趋势会将相同的设置用作缺省配置。

要配置历史趋势

1. 双击趋势，或选定趋势，然后在**特别**菜单上，单击**动画链接**。此时会出现**历史趋势配置**对话框。



提示 如果右击任意历史趋势配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 在**历史标记**框中，输入要用于趋势的标记名。

   如果输入的标记名当前尚未在 “标记名字典” 中定义，则系统会询问是否要立即去定义它。如果选择**是**以立即定义标记名，则 InTouch 将自动显示**标记名字典**对话框，并使用缺省标记名类型**历史趋势**。（标记名必须定义为**历史趋势**型）。每个历史趋势必须使用不同的标记名。

3. 在**初始化时间长度**框中，输入要水平 （x 轴）显示的趋势时间长度，然后选择时间长度的时间增量选项。

   示例：如果在**初始化时间长度**框中输入 30，然后选择**分钟**，则图表的水平时间长度会是 30 分钟。

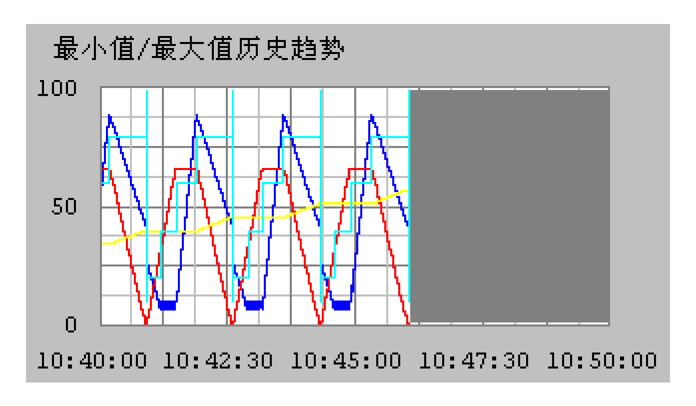
4. 如下选择要用于趋势的**初始化显示模式**：

初始化显示模式	描述
最小 / 最大	图表上的每个象素将显示以象素表示的时间点的最小到最大范围。
平均	显示每个象素的平均值，例如时间段。

5. 在**颜色**组中，单击**图表颜色**框以打开 InTouch 调色板。单击调色板中要用作趋势背景的颜色。
6. 在**颜色**组中，单击**边框颜色**框以打开 InTouch 调色板。单击调色板中要用作趋势边框的颜色。

   对所有的颜色选项重复此过程。

   右边的空白区表示该段时间内没有收集数据，原因是 WindowViewer 没有运行或 “历史记录” 没有激活。



7. 在**时间刻度数**组的**主刻度数**框中，输入趋势中所需的主时间刻度数，然后选择要用于刻度线的颜色。

   主时间刻度数的数目必须是**副 / 主刻度数**的偶数倍。

8. 在**时间刻度数**组的**副 / 主刻度**框中，输入每个主时间刻度内可见的副时间刻度数，然后选择要用于刻度线的颜色。
9. 如果要在趋势的顶部显示时间标签，请在**时间刻度数**组中，选择**顶部标签**。
10. 如果要在趋势的底部显示时间标签，请在**时间刻度数**组中，选择**底部标签**。趋势既可包含顶部标签和底部标签，也可以没有任何标签。

11. 如果使用了时间标签，请在**时间刻度数**组的**副 / 主刻度**框中，输入趋势的每个主时间刻度线的时间标签数。
12. 在**时间刻度数**组中，选择要用于主时间刻度线的颜色。
13. **值刻度数**组中的设置与**时间刻度数**组中的设置采用相同方式进行配置。副值和主值刻度确定了趋势的垂直坐标值 （Y 轴）范围。此范围使用 “工程单位”，并且对于所有要绘制趋势的标记名都是相同的。

提示 要在运行时显示副值和主值刻度的小数点，必须在此处设置它们的格式。例如， 0.00 - 100.00。



14. 在**标记名**框中，通过双击打开**供应器列表**对话框。选择要用于此**笔**的标记名供应器。单击 “确定”。此时会出现一个浏览器，显示该供应器的本地或远程标记名。双击某个标记名以选择它。对于要用每支**笔**绘制其趋势的标记名或表达式，重复此过程。

    在同一个趋势图中，最多有 8 支笔是可见的。（无法记录或绘制消息型标记的趋势）。

备注 FactorySuite Productivity Pack 包括一个可供您用 16 支笔画图的 “笔配置网格”。如需有关详细信息，请参阅 *Productivity Pack 用户指南*。

---

15. 单击颜色框，以选择趋势中由每支笔用来绘制标记名的颜色。
16. 在**宽度**框中，输入每支笔的象素宽度值。

备注 选择的笔宽如果大于 1，则会显著降低屏幕更新和打印时的性能。

17. 如果希望操作员能够在运行时改变趋势的配置，请选择**允许运行时改变**。这些改变包括更改笔的赋值情况、起始日期、时间等。

    如果选择此选项，则在运行期间操作员单击趋势（或者在使用触摸屏时触摸）时，会出现**运行时设置**对话框，供操作员改变趋势。

    如需有关详细信息，请参阅“在运行时更新历史趋势”。

18. 单击**选择显示字体**以访问**字体**对话框，选择要用于趋势显示的字体、样式和大小。

19. 单击**确定**。

使用历史趋势向导

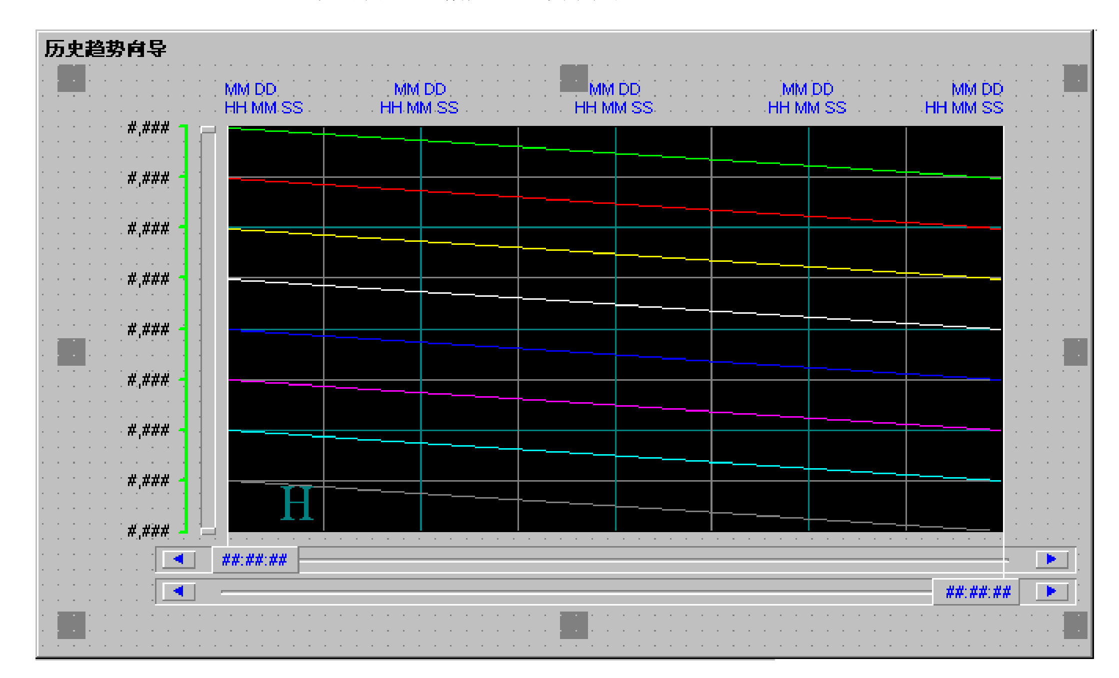
InTouch 提供了一种快捷、简便的方法供您创建历史趋势：趋势向导。通过使用趋势向导，您只需单击几下鼠标，就可以配置一个包含指示器和缩放等在内的一应俱全的历史趋势。

要使用历史趋势向导

1. 单击**向导工具栏**中的向导工具。此时会出现**向导选择**对话框。



2. 在向导列表中选择**趋势**，以显示可用的趋势向导。
3. 选择**含指示器和刻度的历史趋势**向导，然后单击**确定**。此时对话框关闭，您的窗口会再次出现，且光标处于“粘贴”模式。
4. 单击窗口以粘贴该趋势向导。



提示 趋势对象类似于在 WindowMaker 中绘制的任何其它对象。您可以用鼠标抓住趋势对象来移动它，也可以通过抓住对象的一个“手柄”来调整其大小。在窗口中，您可以放置多个趋势图。

5. 现在您就可以开始配置趋势向导了。

6. 双击趋势向导以打开**历史趋势图表向导**配置对话框：

历史趋势图表向导

趋势向导需要两个标记来操作，请在下面键入。
如果下面键入的标记不存在，该向导将创建它们
单击'建议'可获得关于名的建议。

历史趋势：HistTrend1　(历史趋势)

笔尺度：HistTrendPenScale　(内存整型)

笔刻度标记用于显示工程单位.如果您也使用趋势图例向导，请也在
那里指定同种笔尺度标记名.

图表颜色(O)

图表：　边框：　值：

指示器：　时间：

确定(O)　取消(C)　建议(S)　值(V)...　时间(T)...　笔(P)...

提示 如果您右击向导配置对话框中的文本框，则会出现一个菜单，显示可用于所选文本的各条命令。

7. 输入配置趋势所需的信息，然后单击**确定**。

   如果要让向导自动填写配置值，请单击**建议**。给历史趋势向导配置的设置与创建历史趋势对象时所配置的内容相同，这些对象是使用**绘制对象工具栏**中的 WindowMaker 趋势工具所绘制的。

8. 要在趋势中添加缩放和移动功能，或在趋势中添加笔的控件，请分别使用趋势“缩放 / 平移面板”和“趋势笔图例”向导。要让所有这些组件协同工作，它们必须使用**历史趋势**标记名。

   与所有的 InTouch 向导一样，此向导也可以分解成不同的组件。

要分解向导

1. 选择历史趋势向导。
2. 在**排列**菜单上，单击**分解单元**，或者单击**排列工具栏**中的分解单元工具。
3. 随后您就可以根据需要来自定义它。

记录标记名

在 WindowViewer 中，每次发生的变化超过指定的**记录死区**时，要记录的标记名的值会写入历史记录文件，在缺省条件下，无论有无变化，均是每小时写入一次。要将标记名的值写入历史记录文件，必须将它配置为记入“标记名字典”中。

对于整型和实型（浮点）标记名，您可以在它们各自的详细资料对话框中设置**记录死区**。**记录死区**控制标记名的值改变多少个“工程单位”后，才将它记入磁盘。

要配置需记录的标记名

1. 在**特别**菜单上，单击**标记名字典**，或者在“应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。



2. 打开所需标记名的定义，然后选择**记录数据**。

备注 要实际记录标记名，则必须启用记录选项，具体如下节所述。

如果将标记名从记录改为不记录，则会无法访问已经为该标记名记录的数据。

WindowMaker 运行期间，在 WindowViewer 要记录的内容发生任何的改变均会被忽略，直至重新启动 WindowViewer。

“最小 / 最大工程单位”对于显示历史趋势数据而言非常重要。历史趋势显示 0 - 100% 的工程单位范围。

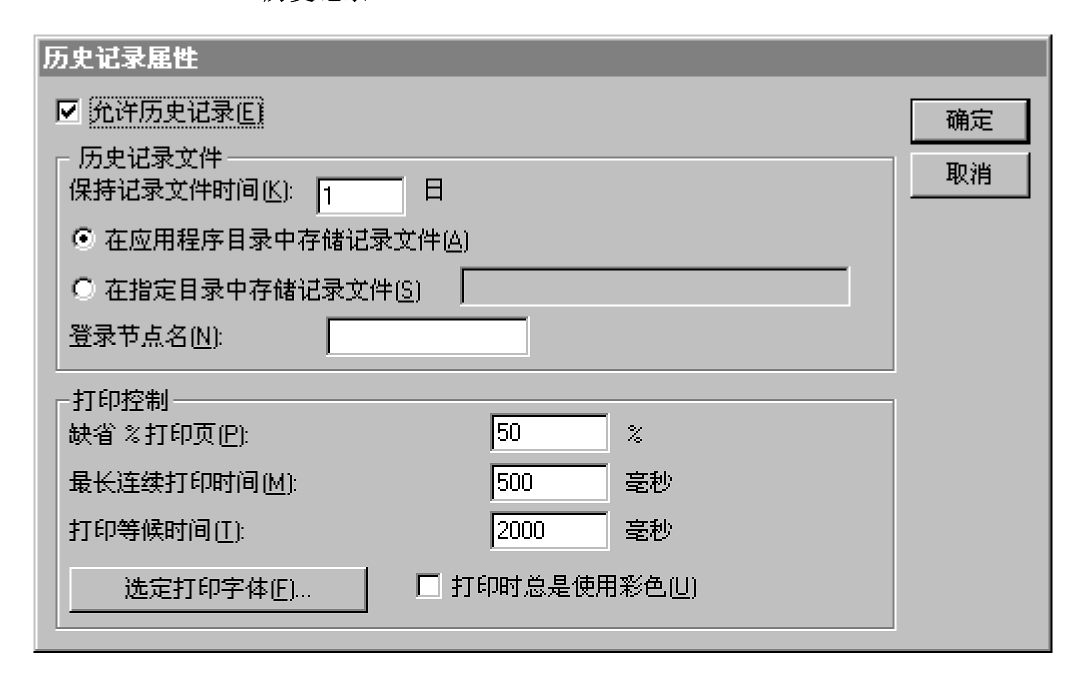
配置历史记录属性

为了将已配置**记录数据**选项的标记名写入历史记录文件，您必须选择全局记录功能。

要配置历史记录

1. 在**特别**菜单上，指向**配置**，然后单击**历史记录**。此时会出现**历史记录属性**对话框。

2. 要快速访问该对话框，请在 “应用程序浏览器” 中的**配置**下，双击**历史记录**。



提示 如果右击任意历史趋势配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

3. 选择**允许历史记录**以打开全局标记名记录功能。

4. 在**保持记录文件时间**框中，输入要将记录文件保存到磁盘的天数（在今天之前）。

   InTouch 会在每天 （24 小时）创建并保存两个历史记录文件。因此，在设置此值时必须考虑磁盘空间。如果硬盘没有足够的可用空间来保存历史记录文件，则记录功能将会停止，此时您必须清理磁盘空间才能重新开始记录。通过将内部标记名 **$HistoricalLogging** 链接到按钮或 QuickScript，或通过使用 WindowViewer 中的**重新启动历史记录**命令，您可以在运行时启动和停止历史记录。

   例如，如果输入 10，而今天是本月的第 12 天，则从 2 号到 12 号（包括今天在内的 10 天）的记录文件将保存到磁盘上。 1 号的文件会被自动删除。如果输入 0，则记录文件将无限期保留。

5. 如果需要将历史记录文件保存到应用程序目录，请选择**在应用程序目录中存储记录文件**。或者，选择**在指定目录中存储记录文件**，并输入要使用的目录的完整路径。

备注 此输入项必须是一个 DOS 路径，如 c:\histlog ；或者，如果执行分布式历史，则它必须是一个 "通用命名惯例"（Universal Naming Convention，简称 UNC）路径，如 \\node\share\directory。

如果配置为将历史数据写入主应用程序节点的 "应用程序目录"，则所有 NAD 节点均会尝试将它们的历史数据写入主应用程序。为避免出现此种情形，需要在每个 NAD 节点上将历史数据配置为写入本地目录，**而不是**主应用程序节点。

如需有关分布式历史的详细信息，请参阅 "分布式历史系统"。

缺省条件下，历史记录文件的命名方式如下：

YYMMDD00.LGH 和 YYMMDD00.IDX

其中：

YY	显示创建文件的年份 (99,01)
MM	显示创建文件的月份 (01-12)
DD	显示创建文件的日期 (01-31)
00	始终显示零

例如，如果文件创建于 1997 年 10 月 31 日，则它们将命名如下：

97103100.LGH

和

97103100.IDX

备注 此版 InTouch 支持扩展名为 .LGH 及 .IDX 的较新记录文件。较早版本的 InTouch 使用给记录文件使用扩展名 .LOG。

6. 在**登录节点名**框中，输入将记入历史记录文件的 NetDDE 节点名（不是计算机名）。
7. 在**缺省 % 打印页**框中，输入页面大小占趋势大小的百分比。

   示例：如果在打印历史趋势时使用 50%，则它会占去半页 （垂直和水平）。此种尺寸的打印输出花费的时间约为整页打印时的 1/4。

提示 影响打印历史趋势性能的因素有很多。主要的因素是打印页上的图表大小。您可以通过减小所用页面的百分比来提高性能。

8. 在**最长连续打印时间**框中，输入历史趋势打印模块连续打印所需的毫秒数 （处理器时间间隔）。

9. 在**打印等候时间**框中，输入历史趋势打印模块在打印输出之间等待的毫秒数。
10. 如果使用彩色打印机或绘图仪，请选择**打印时总是使用彩色**。
11. 单击**选定打印字体**以访问 Windows **字体**对话框。
12. 单击**确定**以保存设置并关闭对话框。

控制历史记录频率

通过**历史记录**命令启用记录功能时，如果标记名的值*没有*发生变化，则指定要记录的所有标记名的值会每隔一小时自动写入“历史记录”文件一次。如果标记名*确实*发生变化，并且工程单位值超过其“记录死区值”（缺省值为 0），则会在变化时自动记录。通过在 **INTOUCH.INI** 文件（位于应用程序目录中）的 **[INTOUCH]** 部分中添加参数 **ForceLogging=#**，可以改写该缺省值。此参数的值用分钟表示，可以设置为 5 和 120 之间的一个值。

要记录标记名的*当前*值（即使变化小于或等于记录死区值），请添加参数 **ForceLogCurrentValue=1**。例如：

通过添加参数 **ForceLogging=15** 和 **ForceLogCurrentValue=1**，当前标记名值会至少每隔 15 分钟或是在标记名值发生变化时写入“历史记录”文件。

备注 ForceLogCurrentValue 参数强制记录标记名的当前值，即使变化小于或等于“记录死区值”。

如需有关“记录死区”的详细信息，请参阅“记录标记名”。

在运行时配置历史趋势

如果配置历史趋势时选择了**允许运行时改变**选项，则该趋势在 WindowViewer 中会是“触控”的，操作员将可以改变笔的赋值情况、改变开始日期和时间等。

要在运行时配置历史趋势

1. 在 WindowViewer 中单击趋势，此时会出现**历史趋势设置**对话框。



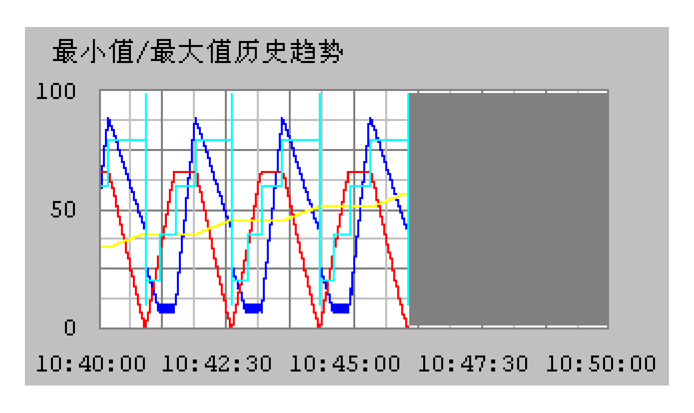
2. 在**图表开始**组中，输入图表的开始日期和时间。
3. 选择图表的**显示模式**。

> **备注** 趋势的显示模式会影响性能。首要因素是生成趋势时所画线条的长度。线越长，生成趋势图所需的时间也就越长。线宽也是影响性能的一个因素，画宽线条的时间要明显长很多。生成**最小 / 最大**或**平均 / 散点**趋势图通常比生成**均值 / 棒图**的速度要快得多。

下面举例说明这三种模式。

最小 / 最大历史趋势

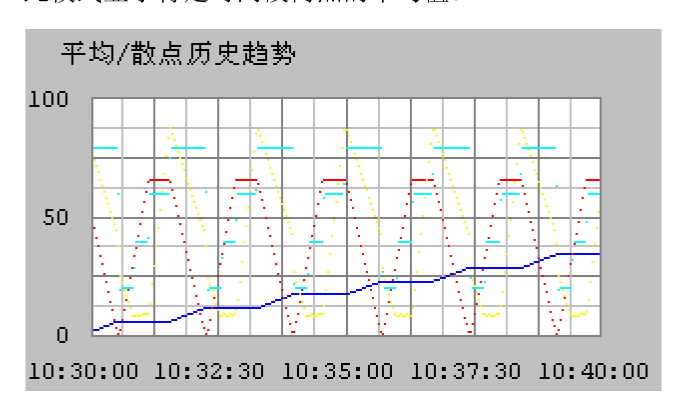
此模式以 “工程单位” 的百分比为纵坐标，以时间间隔为横坐标来显示趋势或变化，并且着重强调时间流和变化率，而非变化量。



备注 右边的空白区表示该段时间内没有收集数据，其原因是 WindowViewer 没有运行或者历史记录未被激活。

平均 / 散点历史趋势

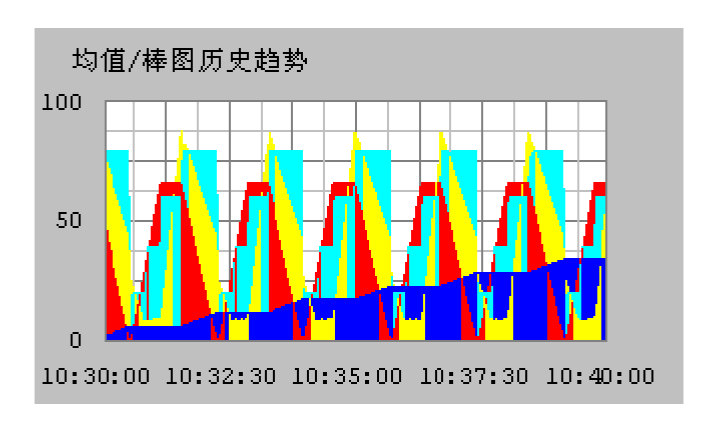
此模式显示特定时间段内点的平均值。



如需有关 “平均 / 散点历史趋势” 的详细信息，请参阅 “平均 / 散点历史趋势计算”。

均值 / 棒图历史趋势

此模式以条形图的形式显示特定时间段内点的平均值。



4. 在**图表长度**框中，输入显示在趋势中的水平 （x 轴）时间长度，然后选择该长度的时间增量。例如，如果输入 1 并选择**小时**，则趋势长度会是 1 小时。
5. 在**图表范围**框中，输入缩放趋势的 “工程单位” 刻度的百分比 （显示在趋势中的 y 轴范围）。

备注 范围的单位是 “工程单位” 刻度的 “百分比”。取值范围应该是 0 到 100。例如，如果要记录选定标记从 40% 到 45% 的变化趋势，请分别在**最小 %** 和**最大 %** 范围框中输入 40 和 45。

---

6. 单击每个**笔 #** 来选择要用该笔绘制其趋势的标记名。此时 “标记浏览器” 会以过滤选择模式出现。



备注　对于选定的标记源，只有在定义时选定了**记录数据**选项的标记名才会显示出来。

---

7. 双击要在趋势上使用选定笔画图的标记名，或选择标记名，然后单击**确定**。此时会再次出现**历史趋势设置**对话框，并显示原先单击的**笔 #** 按钮旁边的选定标记名。

提示　您可以单击**过滤器**箭头来打开已定义的过滤器列表，供您用于填写 “标记浏览器”。此列表的第一项是 < 无 >，表示没有使用过滤器。对于选定的标记源，只有在定义时选定了**记录数据**选项的标记名才会显示。

使用过滤器，或单击**过滤器** 按钮并创建新过滤器时，“标记浏览器” 会重新显示满足所选标记源的过滤器中指定的标准、并且选定了**记录数据**选项的所有标记名。

如需有关 “标记浏览器” 和过滤器的详细信息，请参阅第 6 章 “标记名字典”。

8. 单击**打印**以打印历史趋势。

打印操作可以在 “后台”进行，而 WindowViewer 可以继续处理所有的其它输入。在打印时， WindowViewer 会在菜单中添加两个项目：**取消打印**和 ***X* % 完成**。单击**取消打印**将取消当前的打印作业。

在选择**打印**之，在 WindowViewer 菜单栏中的**取消打印**和 **X % 完成**菜单项消失之前，不要改变趋势。在此期间， WindowViewer 会将趋势信息保存到内存中以便打印。这两个菜单项从菜单栏中消失之后，您就可以改变趋势而不会影响进行中的打印作业。

您可以创建一个按钮并将它链接至执行 **PrintHT** QuickScript 函数的动作 QuickScript，以打印历史趋势。

```
PrintHT(HistTrendTagname);
```

打印操作使用当前的历史趋势作为打印的基础。因此，如果**历史趋势设置**对话框中的任何字段发生变化，**打印**按钮将不会激活。您在**历史趋势设置**对话框中单击**确定**，然后访问它并单击**打印**之前，在设置中所作的改变不会打印出来。

备注 使用**打印**选项或 **PrintHT()** 函数打印 “历史趋势”时，不会打印 y 轴和 x 轴值。要打印 x 轴和 y 轴值，请使用 **PrintWindow()** 或 **PrintScreen()**。

如需有关打印历史趋势的详细信息，请参阅 “配置历史趋势打印”。

平均 / 散点历史趋势计算

“平均 / 散点”是 “历史趋势”对象的三种显示模式之一。标记名定义为启用**记录数据**选项时，该标记名的值会被记入 “历史记录”文件。记录的数据可从 “历史记录”文件读取并显示在 “历史趋势”对象上。在 “平均 / 散点”模式下，显示在对象上的点数取决于标记名值的变化速度。“历史趋势”对象划分成许多刻度。刻度数的限制为 2000，因此，可显示的最多点数是 2000。

例如，考虑某个应用程序，其中要绘制趋势的标记名在一个小时的时间间隔内，每秒记录一次。显示缓冲区中的点数限制为 2000，这表示在 3600 秒的时间范围会被分隔成 3600/2000 或每个 1.8 秒的时间间隔。因此，所画标记名的值在此时间间隔内的实际平均值是 1.8 秒。

如果标记名的值在这 1.8 秒的时间间隔内发生变化，则平均值会被算出，并在 “历史趋势”对象上显示为一个点。如果该值是常数，则它会显示为一条线。

考虑标记名图表长度为 1 秒的另外一种情形。因为显示缓冲区中的点数限制在 2000，1 秒的时间范围会被分割成 1/2000 或每个 0.0005 秒的时间间隔 （即 0.5 毫秒，舍入后为 1 毫秒）。因此，所画标记名的值会是 1 毫秒这个时间间隔内的实际平均值。

将游标与“历史趋势”的图表长度关联且快速移动它时，与慢速移动游标时相比，所显示的点数会有所减少。

在运行时更新历史趋势

在 WindowViewer 中，首次显示历史趋势时，它会显示指定配置的数据。与实时趋势不同，历史趋势自己不会连续更新。要在显示初始数据后让历史趋势自行更新，必须对趋势进行更改。您可以使用下列任何方法来更新趋势：

要在运行时更新历史趋势

1. 在**历史趋势配置**对话框中选择**允许运行时改变**选项（在 WindowMaker 中），使操作员可以手动改变趋势的时间和（或）日期以强制更新。
2. 在 QuickScript 中或在按钮上使用下面的语句，以允许操作员更新图表：
   `Hist_TrendTag.UpdateTrend = 1`
3. 在 QuickScript 中或在按钮上使用下面任何语句：

   `HTUpdateToCurrentTime(Hist_Tag);`

   `HTScrollLeft(Hist_Tag,Percent);`

   `HTScrollRight(Hist_Tag,Percent);`

   `HTZoomIn(Hist_Tag,LockString);`

   `HTZoomOut(Hist_Tag,LockString);`

   `HTSetPenName(Hist_Tag,PenNum,Tagname);`
4. 更改下列任意趋势标记名点域：

   .ChartStart

   .ChartLength

   .MaxRange

   .MinRange

   .Pen1-.Pen8

如需有关使用 QuickScript 函数和点域的详细信息，请参阅 *InTouch 参考指南*。

配置历史趋势打印

影响打印历史趋势性能的因素有很多。最主要的因素是打印页上趋势的大小。趋势的显示模式也会影响打印性能。**最小 / 最大**或**平均 / 散点**打印输出通常要比**平均 / 棒图**趋势的打印输出快得多。趋势上的线条越长、越宽，所需打印时间也越长。

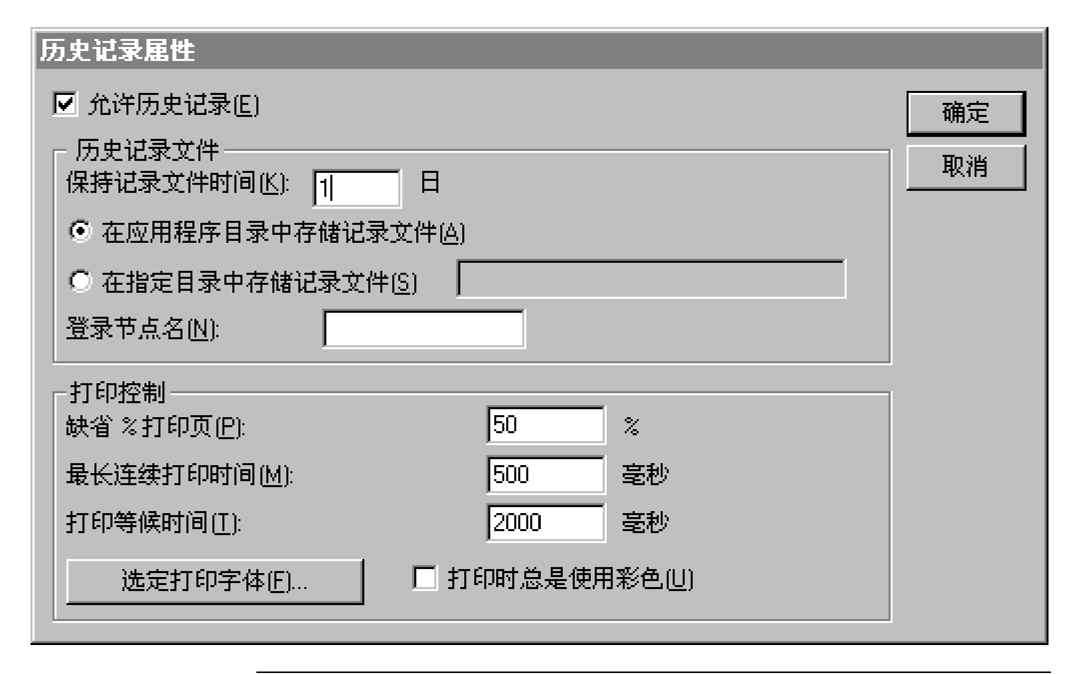
因为打印操作在“后台”进行，InTouch 会给打印进程分配一定的时间，同时会给其它进程分配一定的时间。此方程中的时间由配置历史记录时在**最长连续打印时间**和**打印等候时间**框中设置的值控制。

换句话说，InTouch 会用**最长连续打印时间**框中指定的毫秒数来处理打印，然后用**打印等候时间**字段中指定的毫秒数来处理其它请求。为提高打印优先级，请增大**最长连续打印时间**值，同时减小**打印等候时间**值。要降低打印优先级，则刚好相反。

要配置历史趋势打印

1. 在**特别**菜单上，指向**配置**，然后单击**历史记录**。此时会出现**历史记录属性**对话框。

   提示 要快速访问该对话框，请在“应用程序浏览器”中的**配置**下，双击**历史记录**。



提示 如果右击任意历史趋势配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 在**缺省 % 打印页**框中，指定用于打印趋势的页面百分比。

   如果在此框中输入 50，则 WindowViewer 会使用半页 （垂直和水平）。此尺寸的打印输出所花时间约占整页打印输出的 1/4。

   您可以研究使用 **PrintWindow** QuickScript 函数作为一种替代打印方式。

3. 在**最长连续打印时间**框中，输入打印模块连续打印的处理时间 （毫秒）。

4. 在**打印等候时间**框中，输入打印模块在使用另一处理时间之前等待的时间 （毫秒）。

   影响打印性能的另一个因素是为趋势选定的背景颜色。在大多数情况下，白色背景在打印时要快得多。您最好试一下白色和其它颜色的对比打印，看是否有明显差别。

5. 单击**选定打印字体**以访问**字体**对话框，选择要用于打印输出的字体、样式和大小。

6. 单击**确定**。

历史趋势点域

对于给定的历史趋势标记名，有许多**点域**只适用于历史趋势标记名。下面简要介绍每个历史趋势**点域**。

点域	描述
.ChartLength	可读 / 写整型标记名点域，用于控制显示在“历史趋势”图中的时间长度。 **.ChartLength** 以秒为单位显示图表长度。
.ChartStart	可读 / 写整型标记名点域，用于控制开始时间和 （或）滚动相应的历史趋势。 **.ChartStart** 显示自 70 年 1 月 1 日午夜 12 点以来已过的时间 （以秒为单位）。
.DisplayMode	可读 / 写模拟型标记名点域，用于确定在趋势图上显示值的方式。

点域	描述
.MaxRange, .MinRange	可读 / 写实型标记名点域，用于表示当前绘制趋势的每个标记名所显示的标记名 “工程单位”范围的百分比。 **.MaxRange** 和 **.MinRange** 的限制是从 0 到 100，且 **.MinRange** 总是小于 **.MaxRange**。如果赋给这些字段的某个值小于 0 或大于 100，则这些值会被调整为 0 或 100。如果 **.MinRange** 大于或等于 **.MaxRange**，则该趋势图不会显示任何数据。
.Pen1 - .Pen8	可读 / 写 TagID 型的标记名点域，用于控制每支笔当前正在绘制其历史趋势的标记名。TagID 型标记名只能等于另一 TagID 型标记名。它不能与其它标记名类型混用，除非给其它标记名添加了 **.TagID** 扩展名。 **.TagID** 不能用于远程历史供应器标记名。
.ScooterLockLeft	可读 / 写离散型点域。此点域值为 TRUE 时，“右”指示器不能移到左指示器位置的左边。（0=FALSE、 1=TRUE）。
.ScooterLockRight	可读 / 写离散型点域。此点域值为 TRUE 时，“左”指示器不能移向右指示器位置的右边。（0=FALSE、 1=TRUE）。
.ScooterPosLeft	可读 / 写实型点域，代表左指示器的位置（范围从 0.0 到 1.0）。
.ScooterPosRight	可读 / 写实型点域，代表右指示器的位置（范围从 0.0 到 1.0）。
.TagID	可读 / 写 TagID 标记名点域，与 “历史趋势”.Pen1 - .Pen8 TagID 标记名一起使用，监视和 （或）控制某支笔当前正在绘制其趋势的标记名。
.UpdateCount	趋势的检索完成时会递增的只读整型点域。
.UpdateInProgress	只读离散型点域，显示历史数据的检索状态（0 = 无进行中的检索， 1 = 检索正在进行中）。
.UpdateTrend	可读 / 写离散型标记名点域，可设置为 1 时以使用所有的当前值来更新 “历史趋势”。

如需有关使用**点域**的详细信息，请参阅联机 *Intouch 参考指南*。

历史 QuickScript 函数

您可以使用多个内部函数来指定每支笔绘制其趋势的标记名、显示指示器所在位置的值，以及按百分比滚动趋势等。

如需有关如何使用这些函数及其有效参数的完整示例，请参阅 *InTouch 参考指南*。

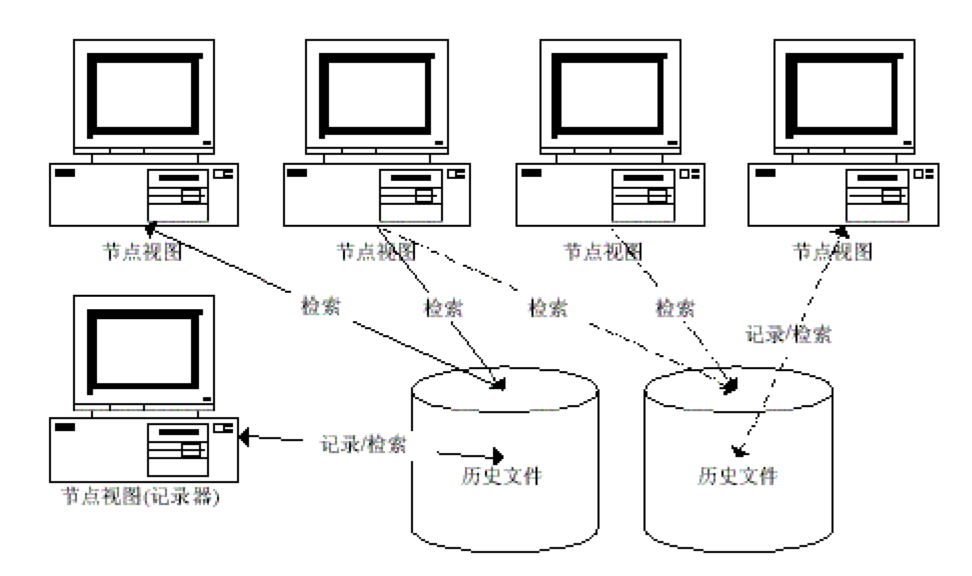
函数	描述
HTGetLastError	确定最后一次检索指定的笔时是否存在错误。
HTGetPenName	返回指定趋势的指定笔号当前所用的标记名。
HTGetTimeAtScooter	返回样本自 1970 年 1 月 1 日 GMT 00:00:00 以来的时间 （以秒为单位），该样本位于由 *ScootNum* 和 *ScootLoc* 指定的指示器位置。 *UpdateCount*、*ScootNum* 及 *ScootLoc* 中的任一个参数发生变化时，便会对表达式重新求值。这可以确保在新的检索或指示器移动之后会对表达式重新求值。
HTGetTimeStringAtScooter	返回包含时间 / 日期的样本字符串，该样本位于 *ScootNum* 和 *ScootLoc* 指定的指示器位置处。*UpdateCount*、*ScootNum* 及 *ScootLoc* 中的任一个参数发生变化时，便会对表达式重新求值。这可以确保在新的检索或指示器移动之后会对表达式重新求值。字符串的格式确定了返回值的内容。
HTGetValue	返回整个趋势的指定笔的请求类型的值。
HTGetValueAtScooter	返回指定的指示器位置、趋势和笔号处的样本的请求类型值。通过使用 *UpdateCount* 参数，可以在检索完成之后对表达式求值。
HTGetValueAtZone	返回某个趋势指定笔的左、右指示器位置之间的数据的请求时间值。
HTScrollLeft	将趋势的起始时间值设置为比当前起始时间早趋势宽度的一定百分比。其效果是将图表的日期 / 时间向左滚动给定的百分比。

函数	描述
HTScrollRight	将趋势的起始时间值设置为比当前起始时间晚趋势宽度的一定百分比。其效果是将图表的日期 / 时间向右滚动给定的百分比。
HTSetPenName	为趋势笔指定不同的标记名。
HTUpdateToCurrentTime	通过将结束时间设置为当前时间来检索并显示数据。开始时间将等于 EndTime（结束时间）减去图表的 “宽度”。
HTZoomIn	计算新图表的宽度和开始时间。如果趋势的 .ScooterPosLeft 为 0.0 而 .ScooterPosRight 为 1.0，则新图表的宽度将等于旧图表的宽度除以 2。新的开始时间将根据 *LockString* 的值进行计算。
HTZoomOut	计算新图表的宽度和开始时间。新图表的宽度等于旧图表宽度乘以 2。新的开始时间将根据 *LockString* 的值进行计算。

分布式历史系统

InTouch 提供的分布式历史系统可从任何 InTouch 应用程序甚至跨网络检索历史数据。此系统通过同时从多个历史数据库远程检索数据，扩展了标准的 InTouch 历史记录系统的功能。这些数据库被称为历史供应器。您最多可同时显示八个历史供应器，且每个供应器对应历史趋势图中的一支笔。

使用分布式历史系统的功能，您可以轻松配置一个网络系统来访问多个历史记录供应器：



每个分布式历史文件限于在一个节点写入（记录到）文件。不过，可查看该文件的 InTouch 节点数或类型并没有什么限制。

只有在 5.6 或更高版本的 InTouch 中开发的应用程序才可以成为历史供应器。要远程查看较早版本的历史文件，首先必须将该应用程序转换成 5.6 版或更高版本。

从历史文件检索数据的远程节点可能看不到最近一个小时的数据（基于记录器节点的时间）。远程趋势只可查看已写入记录节点磁盘的数据。

为该标记名采集 22 个样本之后，选定“记录数据”选项的每个标记名的数据会自动写入磁盘。如果执行 **HTUpdateToCurrentTime()** 函数，则无论采集了多少样本，数据均会写入磁盘。缺省条件下，数据每小时会写入磁盘一次。您可以通过在 intouch.ini 文件中添加下面一行代码来改变此时间间隔：

```
ForceLogging=X;
```

其中 **X** 是分钟数，可设置为介于 5 和 120 的任意时间间隔。

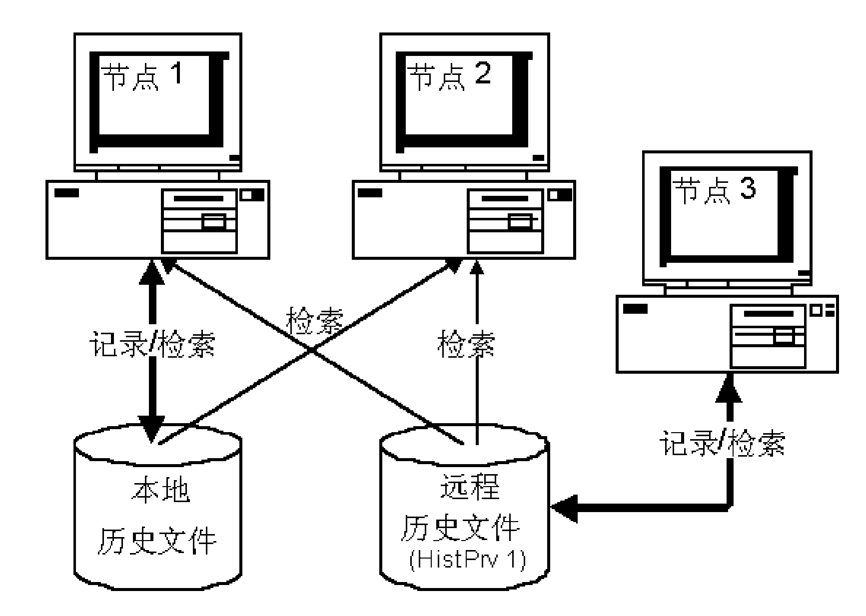
备注 使用“分布式历史”时， Wonderware NetDDE Helper 服务必须正在运行。

如需有关 Windows NT 服务的详细信息，请参阅附录 A “InTouch Windows NT 服务综述”。

使用分布式历史系统

下图说明如何设置分布式历史系统。此系统是使用 “网络应用程序开发”（Network Application Development，简称 NAD）来发布应用程序的典型分布式应用程序。

如需有关 NAD 的详细信息，请参阅第 5 章 “建立分布式应用程序”。



“节点 1”和 “节点 2”包含相同 InTouch 应用程序的备份，不过，该应用程序被配置为只允许 “节点 1”记录到本地历史文件，而这两个节点均可从本地历史文件或远程历史文件检索数据。“节点 3”也可以记录到远程历史文件位置，并可从中进行检索。此供应器 （节点 3）被赋予的名称是 **HistPrv1**。“节点 1”既是开发工作站，也是运行工作站，而 “节点 2”只是运行工作站。

创建此应用程序所需的主要步骤包括：

1. 创建历史供应器列表。
2. 创建并配置历史趋势对象。
3. 配置应用程序的分布式记录功能。
4. 分发应用程序。

所有这些步骤均将在本章中向您介绍。

分发应用程序

您可以手动或使用 NAD 系统来分发应用程序。分发应用程序时，历史供应器列表文件将作为应用程序的一部分进行分发。

如需有关使用 NAD 的详细信息，请参阅第 5 章 “建立分布式应用程序”。

在分发应用程序之后，您可以运行 View 节点，并检索本地标记名和远程历史供应器的标记名。尽管应用程序可以在所有的 View 节点上运行，只有记录节点才会将数据记录到历史记录文件，而其它节点则只能从中读取。

配置分布式历史供应器列表

您要从中检索历史数据的每个远程历史供应器均必须在 InTouch 历史供应器列表中注册。此列表可供指定每个历史供应器的名称和网络位置。每次在 InTouch 中引用历史供应器时，均须使用这些名称。

要配置历史供应器列表

1. 在**特别**菜单上，指向**配置**，然后单击**名称管理器**。此时会出现**分布式名称管理器**对话框。

提示 要快速访问该对话框，请在 “应用程序浏览器”中的**配置**下，双击**分布式名称管理器**。



提示 如果右击任意历史趋势配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

2. 单击**分布式历史**标签，以激活分布式历史供应器属性页。

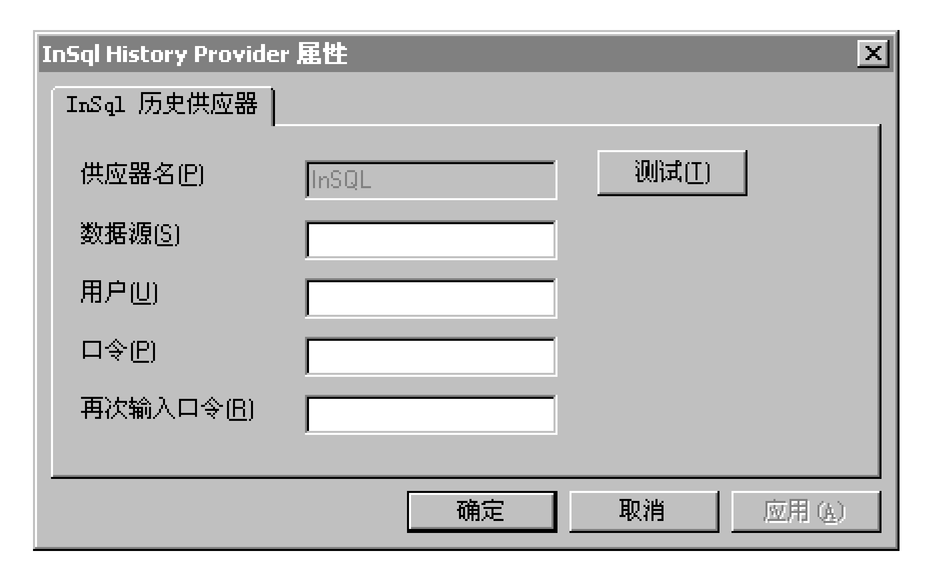
3. 在**供应器名**框中，输入要用于新历史供应器的名称。
4. 如果要访问 InTouch 应用程序的日志文件，请选择 **InTouch 供应器**并在 **UNC** 框中输入 InTouch 应用程序目录的 UNC （Universal Naming Convention，中译 “通用命名惯例”）路径。有效格式为：

   ```
   \\Node\Share\ApplicationPath
   ```

   在上例中， **HistPrv1** 的路径包括节点名 "HistNode"、共享 "C$" 以及应用程序路径 "\Apps\HistApp"。

   如果该 UNC 位置是口令保护的，则必须首先使用 Windows “资源管理器”来建立连接。

   如需有关 UNC 路径的详细信息，请参阅第 5 章 “建立分布式应用程序”。
5. 选择 **InSQL 供应器**以访问 IndustrialSQL Server 运行时数据库中的数据，然后单击**配置 InSQL 供应器**。此时会出现 **InSql History Provider 属性**对话框。



   缺省条件下，最近一次成功登录的登录参数会显示出来。如有必要，请修改登录参数以连接到所选的 IndustrialSQL Server。
6. 在**供应器名**框中，输入要用作 InSQL 供应器的名称 (用户定义的)。
7. 在**数据源**框中，输入 IndustrialSQL Server 数据库所在节点的名称（不超过 35 个字符）。
8. 在**用户**框中，输入登录帐户的名称。
9. 在**口令**框中，输入登录帐户的口令。
10. 在**再次输入口令**框中，再次输入口令以进行确认。
11. 单击**测试**来验证与 InSQL Server 的连接。此时会出现一个消息框，通知您连接是成功还是失败。单击**确定**以关闭该消息框。

12. 单击**确定**。

备注 用户帐户由用户名和口令组成。用户帐户必须与检索数据的权限关联，否则登录会失败。如需有关用户账户的详细信息，请与系统管理员联系。

在 InSQL 数据库中查询 InTouch 趋势对象的数据时，指定时间周期（一行计数）的 1000 个等距的行会被检索出来，并绘制到历史趋势图上。历史趋势上显示的标记名的最小和最大值不见得一定是该标记名实际的最小和最大值。

HistData **不能**从 “InSQL 供应器”检索历史信息。

如需有关历史供应器的详细信息，请参阅 “配置分布式历史供应器列表”。

13. 单击**添加**。

WindowViewer 显示历史供应器名时，历史系统会在供应器列表中查找该名称。如果名称在列表中，则它会从该供应器读取历史记录文件。如果该名称不存在，则引用会被忽略，并在 Wonderware Logger 中写入一则错误消息。此时尽管本地 InTouch 应用程序会被视为历史供应器，它不必在此文件中进行配置。

配置远程历史供应器

历史趋势支持本地和远程历史供应器标记名的显示。

要显示远程历史供应器的标记名

1. 双击历史趋势，以访问**历史趋势配置**对话框。
2. 在每支笔的**标记名**框中，按下面的格式输入标记名：

   HistPrv1.tagname

   每支笔均可引用不同的远程历史供应器。例如，在配置历史趋势时，如果要让 Pen1 绘制在被定义为 HistPrv1 的远程历史供应器中的标记名 Boiler1，则您会在 Pen1 的 “标记名”框中输入 HistPrv1.Boiler1。

备注 .TagID 标记名点域不能用在远程历史供应器标记名的引用中。

使用标记浏览器访问远程历史供应器

下面的操作程序介绍应用程序开发人员可如何使用 “标记浏览器”来选择远程标记名引用。

要将远程历史供应器定义为标记源

1. 创建一个 “访问名”，指定历史供应器所在的**节点名**。

   在 “访问名” 中指定的**节点名**不见得一定是标记名所在节点的实际名称。但是，您必须创建 “访问名”，否则会无法将远程历史供应器定义为标记源。

2. 双击历史趋势。此时会出现**历史趋势配置**对话框。
3. 双击某支笔的**标记名**输入框。此时会出现 “标记浏览器”。
4. 单击 “定义标记源” 按钮 [...] 以便将远程历史供应器定义为标记源。
5. 单击**标记源**箭头，从列表中选择新的远程历史供应器标记源，或单击 “树形视图” 按钮，并在树形视图窗格中选择标记源。随后 “标记浏览器” 会重新显示所选远程历史供应器的标记名。
6. 双击要指定给历史笔的标记名，或者选择它然后单击**确定**。
7. 此时会再次出现**历史趋势配置**对话框，所选标记名会出现在该笔的**标记名**框中，格式如下：**AccessName:Item**。
8. 将 **AccessName:** 部分替换为在 “分布式名称管理器” 中定义的历史供应器名。例如， **HistPrv1.Tagname**。

   此过程看似麻烦，但是，在 “标记浏览器” 中将历史供应器定义为标记源之后，每次双击另一个标记名输入框时，您只需在 “标记浏览器” 中双击标记名，然后将 **AccessName:** 部分替换为历史供应器名就可以了。通过采用此操作过程，在指定远程历史供应器标记名时，您会降低自己的出错几率。

备注 在 WindowViewer 中，如果允许运行时改变历史趋势，则用户单击笔按钮以改变标记名时，会出现 “标记浏览器”，但只能访问本地应用程序的标记名。

如需有关使用 “标记浏览器” 的详细信息，请参阅第 6 章 “标记名字典”。

动态配置远程历史供应器

在运行时，您晐可以动态配置历史趋势的远程历史供应器，为此，您要创建一个 QuickScript，在 **HTSetPenName** 函数中指定远程历史供应器标记名引用。例如：

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

其中， 1 是指用于绘制指定的远程历史供应器标记名的笔。

备注 在运行时，远程历史供应器不支持 “历史趋势设置”对话框和 .Pen。

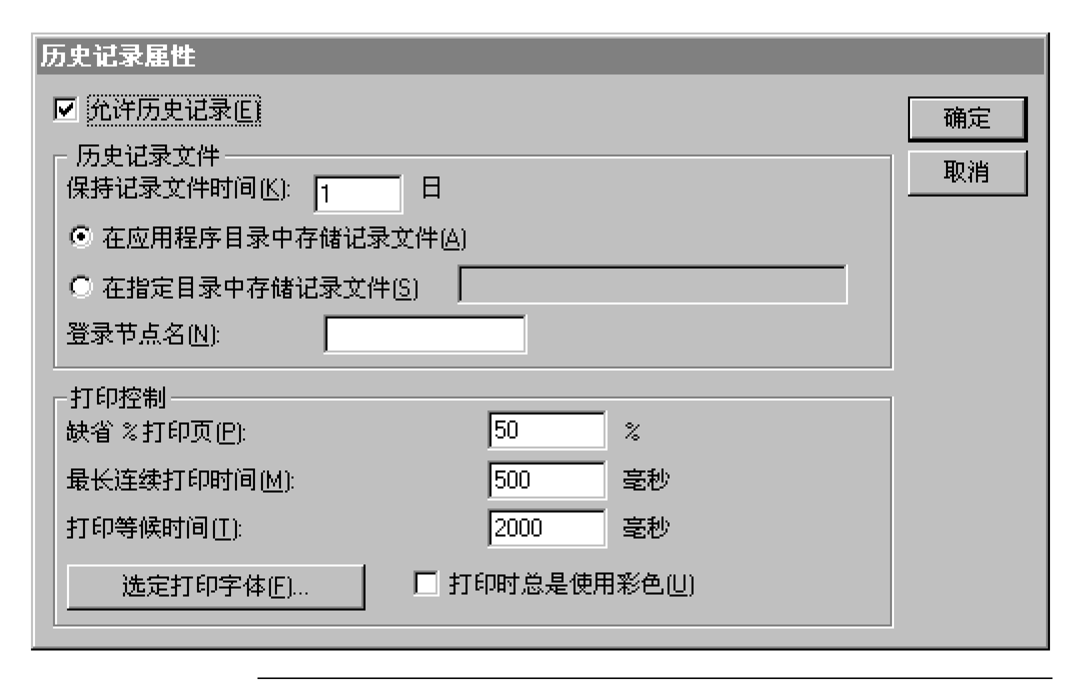
如需有关 InTouch QuickScript 函数的详细信息，请参阅联机 *InTouch 参考手册*。

配置分布式历史记录

要配置分布式历史记录

1. 在**特别**菜单上，指向**配置**，然后单击**历史记录**。此时会出现**历史记录属性**对话框。

   提示 要快速访问该对话框，请在 “应用程序浏览器”中的**配置**下，双击**历史记录**。



提示 如果右击任意历史趋势配置对话框中的文本框，则会出现一个菜单，显示可应用于所选文本的各条命令。

---

2. 选择**允许历史记录**以打开全局标记名记录功能。
3. 选择**在指定目录中存储记录文件**，然后在输入框中输入存储记录文件的位置的路径。您必须输入有效的 “通用命名惯例”（Universal Naming Convention，简称 UNC）路径。例如， "\\Node\Share\Path"。

   如果使用 NAD，则必须确保路径指向与应用程序目录不同的某个目录。

4. 在**登录节点名**框中，输入要记入历史记录文件的节点的名称。

   此设置只允许将在此输入的节点记入历史文件。

5. 单击**确定**。

备注 在将选定了**允许历史记录**选项的应用程序分发到 WindowViewer 节点时，该节点会检查此选项以确定是否应该记录它。如果选定了**允许历史记录**，则可能的设置是：
此域等于 "节点"名 - 允许记录
此域不等于 "节点"名 - 不允许记录

创建历史趋势指示器

指示器是时间刻度上的位置指示器，可改变位置以检索某个精确的时间点的特定数据。通过在指示器**点域**中输入游标对象，您可以在历史趋势显示图上滑动并访问所需的数据段。您可以使用 QuickScript 函数来访问特定指示器位置的平均值、最小值和最大值。通过创建左、右指示器并使用附加的 InTouch QuickScript 函数，可以根据指示器之间或指示器位置上的数据执行分析以返回值。分析类型包括 "平均"、"最小"、"最大"、"最小 / 最大值"、"最小 / 最大工程单位" 及 "标准偏差"。您也可以在两个指示器之间执行缩放操作。

如需有关指示器点域的详细信息，请参阅 "历史趋势点域"。

此外，您还可以添加根据图表上的已知位置来显示数据的功能。在趋势图上进行缩放也很有裨益。以下几页介绍如何使用链接和表达式，以便在历史趋势上采用此项功能。

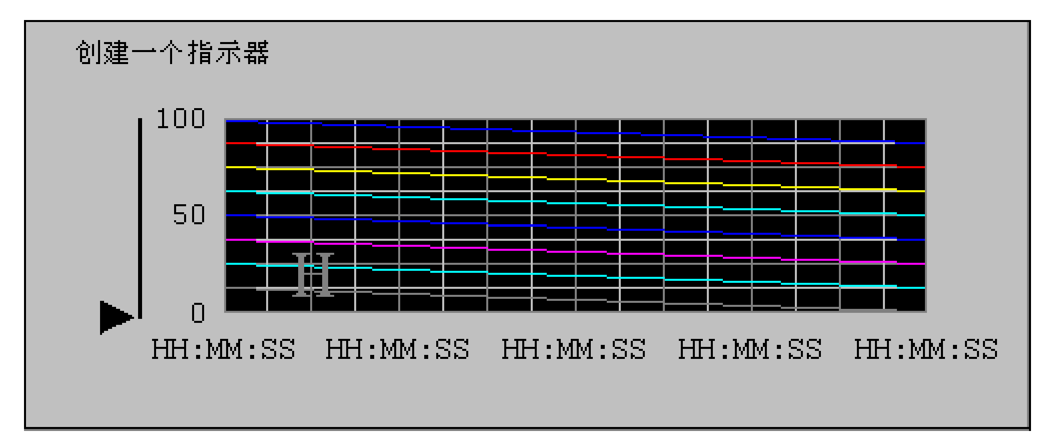
提示 这些功能已经在 "历史趋势向导" 中配置过。

如需有关详细信息，请参阅 "使用历史趋势向导"。

您可以使用**水平游标**链接来创建一个可以在历史趋势顶部移动的指示器。（每个趋势支持使用左、右两个指示器）。

要创建指示器

1. 创建要用作指示器的对象。在下例中，我们将使用多边形和垂直线符号：



2. 要正确定义指示器的水平游标链接，您需要知道图表的宽度。要确定图表的宽度，请从一端到另一端绘制一条水平线，并请注意 WindowMaker 状态栏中显示的值。观察从左边起的第三个数值，这就是您图表的宽度。记下这个数字。删除该“测量”线。
3. 双击指示器对象。此时会出现动画链接选择对话框。
4. 在**触动链接 - 游标**区，单击**水平**。此时会出现**水平游标**对话框。



5. 在**标记名**框中，输入历史趋势名，加上 **.field** 和 **.ScooterPosLeft**。

   例如，如果趋势包含标记名 **Trend1**，则标记名是 **Trend1.ScooterPosLeft**。**在左端**值是 0.0，而**在右端**的值是 1.0。**水平移动向左**的值为 0，而**向右**的值是从趋势一端绘制水平线到另一端时的象素值。（在上例中，图表为 250 个象素宽）。

6. 单击**确定**。

将左指示器移至趋势的最左缘。这样左指示器便设置完毕。对于右指示器，请使用 **.ScooterPosRight** 重复此过程。值域将保持不变。

显示指示器位置的数值

要根据指示器的当前位置来显示值，就需要创建一个数字文本对象，例如，创建数字文本对象 #.00 并使用表达式将其指定给**值显示 - 模拟**链接。例如：

```
HTGetValueAtScooter( "Trend1", Trend1.UpdateCount, 1,
   Trend1.ScooterPosLeft, 1, "PenValue" )
```

此示例在 Trend1 左边的指示器位置检索 Pen1 的值。如果游标在 “运行时”移动，则**值显示 - 模拟**链接将随着在新指示器位置处出现新值而自动更新。

检索区域之间的值

要检索指示器当前位置之间的数据，无论是最大值、最小值、平均值或标准偏差，请创建一个数值文本对象 （如 #.00），并使用表达式将它赋给一个**值显示 - 模拟**链接。例如：

```
HTGetValueAtZone( "Trend1", Trend1.UpdateCount,
   Trend1.ScooterPosLeft, Trend1.ScooterPosRight, 1,
   "PenMaxValue" )
```

本示例将检索 Trend1 上左指示器位置和右指示器位置之间 Pen1 笔的 “最大”值。

缩放

要放大趋势图，请创建一个按钮，并将它指定给**触动按钮 - 动作**链接。选择**按下时**条件，并输入下列 QuickScript：

```
HTZoomIn( "Trend1", "Center" );
```

本示例将以趋势的中央时间为 “基准”，在指示器位于最左和最右位置时，将新图表的宽度设置为等于旧图表的宽度除以 2。如果指示器移动，则新图表的宽度是左、右指示器之间的时间，而不使用 ***LockString*** （“中心”）。

如需有关使用历史 QuickScrip 函数的详细信息，请参阅 “历史 QuickScript 函数”。

历史趋势和夏令时

InTouch 的 “历史记录”功能会根据计算机时钟，在每天午夜创建一个新的 “历史记录”文件。选定历史记录选项的标记名的值发生的变化超过 “记录死区”时，该值将写入 “历史记录文件”。“历史记录”文件中的每条记录均包含该标记名的最新值，以及更新数值的时间和日期。InTouch 历史引擎使用 “通用协调时间”（Universal Coordinated Time，简称 UTC），也称为 “格林尼治标准时间” (Greenwich Mean Time，简称 GMT）作为记录和检索数据的时间。下例介绍 “历史”引擎如何处理 “时区”和 “夏令时”问题：

假定有一个标记名 (**tag1**) 会在 GMT 西部时间上午 10:00:00 时 （计算机时间）改变它的值。

Logged_TimeStamp=Computer_Time+(Offset -DST)

其中， **Logged_TimeStamp** 是在日志文件中输入的记录的 “时间标签”。

示例 **1**

“时区”设置为 “太平洋标准时间” (GMT - 8) 并启用 “夏令时”：

Logged_TimeStamp=10:00:00 +(8-1)=17:00:00

示例 **2**

“时区”设置为 “太平洋标准时间” (GMT -8) 并禁用 “夏令时”：

Logged_TimeStamp=10:00:00 +(8-0)=18:00:00

示例 **3**

“时区”设置 “阿姆斯特丹、柏林、伯尔尼、罗马时间” (GMT +1) 并启用 “夏令时”：

Logged_TimeStamp=10:00:00 +(-1+1)=10:00:00

示例 **4**

“时区”设置为 “蒙罗维亚、卡萨布兰卡时间” (GMT0) 并禁用 “夏令时”：

Logged_TimeStamp=10:00:00 +(-0 +0)=10:00:00

在示例 4 中，日志文件中的 **TimeStamp** 与计算机时间相同 （这是理想设置）。

“历史趋势”也会考虑检索历史数据的计算机的 “时区”和 “夏令时”设置。如果记录计算机与检索计算机之间 “时区”设置不同，在 10:00:00 时 （记录节点时间）写下的记录可能会在检索节点上显示不同的时间。在大多数情况下，如果在相同的计算机 （或在具有相同 “时区”设置的计算机）上执行记录和检索，则所检索的记录是正确的。

要在 Windows 控制面板中设置时区

1. 打开 Windows “控制面板”。
2. 双击**日期 / 时间**图标，或双击 Windows **任务栏**中的时钟。此时会出现**日期 / 时间属性**对话框。



3. 单击**时区**标签，然后单击箭头以打开时区列表。
4. 从列表中选择您的时区。
5. 取消选择**根据夏令时自动调整时钟**。
6. 单击**确定**。

备注 Windows 2000 和 Windows NT 操作系统可能会设置为根据夏令时自动调整时钟。建议您禁用此项功能，因为它会在夏令时切换时导致报告错误的值。要禁用此项功能，请使用 Windows “控制面板”上的“日期 / 时间”工具进行设置，或双击 Windows “任务栏”上的时钟。

如需有关系统时间的详细信息，请参阅第 5 章 “建立分布式应用程序”。

自动改变系统时间

Windows 2000 和 Windows NT 操作系统会试图根据夏令时自动调整时钟。您可以通过使用 Windows “控制面板”中的 “日期 / 时间”工具关闭此项功能，然后手动调整时间，或使用 InTouch QuickScript 在以下这些时间自动设置时钟：

要在春季向前调拨 Windows 2000 和 Windows NT 的时钟

创建下面的 “条件” QuickScript：

```
$Year == yyyy and $Month == 04 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 0 ;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

为真时：

```
DaylightSavingsTime = 1;
```

StartApp "c:\Winnt\System32\control.exe Date/Time";

{ 您可能需要调整路径来反映文件在系统中的位置 }**SendKeys "%(t)" ;**

```
SendKeys "%(03)" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys " {ENTER}"
Flag=O;
```

备注 根据计算机系统的速度，第二个条件脚本 （为真时）产生的延迟可以增大或减小。引入此延迟是为了在发送键之前启动 CONTROL.EXE。

要使用命令行在春季向前调拨 Windows 95 （或更高版本）的时钟

创建下面的 “条件” QuickScript：

```
$Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02
    and DaylightSavingsTime == 0;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 该年时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

为真时：

```
DaylightSavingsTime = 1;
StartApp "c:\Command.com /c time 03:00";
```

要使用 WWDosCommand() 函数在春季向前调拨 Windows NT 时钟

创建下面的“条件” QuickScript：

```
$Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02
   and DaylightSavingsTime == 0;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 该年时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

为真时：

```
DaylightSavingsTime = 1;
WWDosCommand ("time 03:00", "Invisible";
```

要在秋季回拨 Windows 2000 或 Windows NT 操作系统的时钟

创建下面的“条件” QuickScript：

```
$Year == yyyy and $Month == 10 and $Day == dd and
$Hour == 02 and DaylightSavingsTime == 0 ;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 该年时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

备注 每当系统时钟回拨时，历史记录引擎可能会改写历史记录文件中现有的数据。为防止由此造成数据丢失，请在回拨时钟之前备份日志文件。

为真时：

```
DaylightSavingsTime = 1;
StartApp "c:\windows\control.exe Date/Time" ;
```

备注 您可能需要改变路径来反映 CONTROL.EXE 在计算机系统中的位置。

Flag 是一个用户自定义的内存整型标记名。它的初始值为 0。

```
Flag==1;
```

每 1000 毫秒为真时：

```
SendKeys "%(t)" ;
SendKeys "%(01)" ;
SendKeys "01" ;
SendKeys "%{TAB}" ;
```

```
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "% {ENTER}"
Flag=1;
Flag==1;
```

备注 根据计算机系统的速度，第二个条件脚本 （为真时）产生的延迟可以增大或减小。引入此延迟是为了在发送键之前启动 CONTROL.EXE。

要使用命令行在秋季回拨 Windows 95 的时钟

创建下面的 “条件” QuickScript：

```
$Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02
    and DaylightSavingsTime == 0;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 该年时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

为真时：

```
DaylightSavingsTime = 1;
StartApp "c:\Command.com /c time 01:00";
```

要使用 WWDosCommand() 函数在秋季回拨 Windows NT 的时钟

创建下面的 “条件” QuickScript：

```
$Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02
    and DaylightSavingsTime == 0;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 该年时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

为真时：

```
DaylightSavingsTime = 1;
WWDosCommand ("time 01:00","Invisible";
```

备注 每次回拨系统时钟时，历史记录引擎可能会改写历史记录文件中现有的数据。为防止由此造成数据丢失，请在回拨时钟之前备份日志文件。

要在秋季回拨时钟，同时防止丢失数据

创建下面的 “条件” QuickScript：

```
$Year==yyyy and $Month == 04 and $Day == dd and $Hour ==02
   and DaylightSavingsTime == 0;
```

其中：

yyyy = 年份 （如 1993、 1994 或 1995 ...）

dd = 该年时间变化的日期

DaylightSavingsTime = 用户自定义的内存离散型标记名，表示夏令时。它的初始值为 0。

为真时：

```
DaylightSavingsTime = 1;
```

```
$Historicallogging=0;
```

此脚本将停止历史记录功能。

```
StartApp "C:\Winnt\system32\control.exe date/time
```

备注 您可能需要改变路径来反映 CONTROL.EXE 在计算机系统中的位置。

Flag 是一个用户自定义的内存整型标记名。它的初始值为 0。

```
Flag==1;
```

每 1000 毫秒为真时：

```
SendKeys "(t)";
SendKeys "01";
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "%{TAB}" ;
SendKeys "{ENTER}"
FileMove("c:\history\9910dd00.igh",
   "c:\backup","monitor");
Flag=2;  Flag==2
```

每 5000 毫秒为 TRUE：

```
FileMove("c:\history\9910dd00.idx", "c:\backup",
   "monitor");
Flag=3;
Flag==3
```

每 5000 毫秒为真时：

```
$Historicallogging=1;
Flag=0;
```

其中：

history 是日志文件所在的目录。

backup 是日志文件要移动到其中的目录。

9910DD.IGH 和 9910DD00.IDX 是要移动的日志文件。

'DD' 将替换为当前日期。

根据日志文件的大小和计算机系统的速度，“条件” QuickScript 中 5000 毫秒的延迟时间可以增大或减小。

HistData 实用程序

通过使用 HistData 实用程序，可采用 DDE （Dynamic Data Exchange，中译 “动态数据交换”）技术访问 InTouch 创建的历史数据文件。它用于将所选的历史数据移入提出请求的程序，例如 Microsoft Excel。通过使用 HistData，您可以即时查看历史数据，或创建文件供日后访问。历史数据的访问可通过在提出请求的程序中使用宏函数或直接在 InTouch 中实现。

备注 HistData 程序应先于使用它的所有程序启动 （然后缩小为一个图标）。

HistData 程序不支持远程标记名引用。

HistData 数据库

HistData 程序包含自己的内部数据库。内部数据库中的项目用于指定起始周期、持续时间及采样间隔等，以便访问历史数据。下表列出 HistData 程序中定义的各个项目：

DATADIR

消息型：包含历史数据文件的目录的路径名，例如 C:\InTouch\App。

DBDIR

消息型：包含 “InTouch 标记名字典”的目录的路径名，例如 C:\InTouch\App。

STARTDATE

消息型：MM/DD/YY 格式的数据采样开始日期

STARTTIME

消息型：使用 24 小时时钟的 HH:MM:SS 格式的数据采样开始时间。

DURATION

消息型：返回数据的时间长度。 **DURATION** 可以用星期、天、时、分、秒来表示。

下面是有效的字符：
w （星期）、 **d** （天）、 **h** （时）、 **m** （分）、 **s** （秒）。

小数值也允许使用。
例如， .5s 表示 500 毫秒。

若要求提供一个样本，请将 **DURATION** 设置为 0 （零）。

INTERVAL

消息型：采样间隔的时间长度。 **INTERVAL** 可以用星期、天、小时、分以及秒表示，例如 1W 表示 1 个星期。小数值也允许使用，例如用 .25d 表示 6 个小时。

（有效字符与 **DURATION** 的相同）。

备注 DURATION 与 **INTERVAL** 允许使用的最大时间长度均为 6 个星期。这适用于所有的请求类型、天、秒等。例如，如果使用天，则最大值为 42 （7 天 x 6 周 = 42）。

TAGS

消息型：要为其返回历史数据的标记名的列表。 **TAGS** 以 "TagA,TagB,TagZ" 的形式输入。此外，采样日期和 （或）时间可通过使用内部系统标记名 **$Date** 和 **$Time** 来检索。例如：

"$Date,TagA,TagB" 或
"$Time,TagA,TagB" 或
"$Date,$Time,TagA,TagB"

TAGS1, TAGS2,....

消息型：**TAGS** 字符串在 WindowViewer 中最长可达 131 个字符，在 Excel 中最长可达 255 个字符。该字符串可以通过追加标记名项目 "Tags1"、 "Tags2" 等来延长，以满足要求使用更长字符串的需要。如果标记名需要追加额外的标记名文本，请在字符串末尾输入一个加号 (+)。例如：

TAGS="$Date,ProdLevel,ProdTemp,+"

TAGS1="ReactLevel,Temp,GasLevel,+"

TAGS2="MotorStatus"

备注 不允许使用重复标记名，每个标记字符串的最大长度为 512 字节。

PRINTTAGNAMES

离散型：此项目的缺省值为 1，可让 HistData 在输出文件第一行、关联值列的上方打印标记名。如果不打印标记名，则必须将此项目的值更改为 0 （零）。

DATA

消息型：此项目用于以逗号分隔变量格式保存 HistData 程序中的请求数据。它可用通过 DDE 来 **ADVISE** （提示）或 **REQUEST** （请求）数据的其它应用程序使用。

SENDDATA

整数型：设置为 1 时， HistData 会用请求的数据更新 **DATA** 项。更新完成后， **SENDDATA** 会自动重置为 0 （零）。

备注 如果您在使用 **SENDDATA** 时收到一则错误消息，告诉您请求的数据太多，则请缩短 **DURATION** 或减少请求的标记名数目。

FILENAME

消息型：用于写入请求数据的文件的完整路径名，例如 C:\INTOUCH\HDFILE.csv。

WRITEFILE

整数型：设置为 1 时， HistData 会将请求数据写入 **FILENAME** 项目名指定的文件。文件更新完成后， **WRITEFILE** 会自动重置为 0 （零）。

STATUS

离散型：显示最近一次操作的状态。 1 表示成功， 0 （零）表示出错。

ERROR

消息型：包含最近一个错误的说明的字符串。 **STATUS** 为 1 时，它会是“无”； **STATUS** 为 0 （零）时，则它会包含错误消息字符串。

在 InTouch 中使用 HistData

在本节我们已创建一个示例窗口，向您演示在 InTouch 中读取和显示历史数据的方法。

为了让 InTouch 能够从 HistData 程序请求数据，我们定义了下面一个“访问名”：

添加访问名

访问名(M): ViewStream1

节点名(N):

应用程序名(A): HistData

主题名(T): ViewStream1

使用哪个协议 (P)

DDE SuiteLink

要对服务程序提示时(W)

提示所有项 只提示激活项

确定

取消

访问名可以是最长达 32 个字符的任意名称。**主题名**也可以是最长达 32 个字符的任意名称。建议给这两个项目使用相同的名称。**应用程序名**必须是程序名 **HistData** （不含 .exe）。

此外，建议每次使用 HistData 时均选定**提示所有项**选项。

如需有关 “访问名”的详细信息，请参阅第 13 章 “I/O 通讯”。。

在定义 “访问名”之后，为每个 HistData 内部数据库项目创建下面的 I/O 型标记名：

标记名	I/O 标记类型	标记名	I/O 标记类型
DATA	消息	**SENDDATA**	整型
DATADIR	消息	**STARTDATE**	消息
DBDIR	消息	**STARTTIME**	消息
DURATION	消息	**STATUS**	离散
ERROR	消息	**TAGS**	消息
FILENAME	消息	**WRITEFILE**	整型
INTERVAL	消息	**PRINTTAGNAMES**	离散

如果您不仅要将数据发送至 .CSV 文件，而且还要发送至名称为 **Data** 的项，以便可以从其它应用程序访问它，则请创建下面两个额外的标记名：

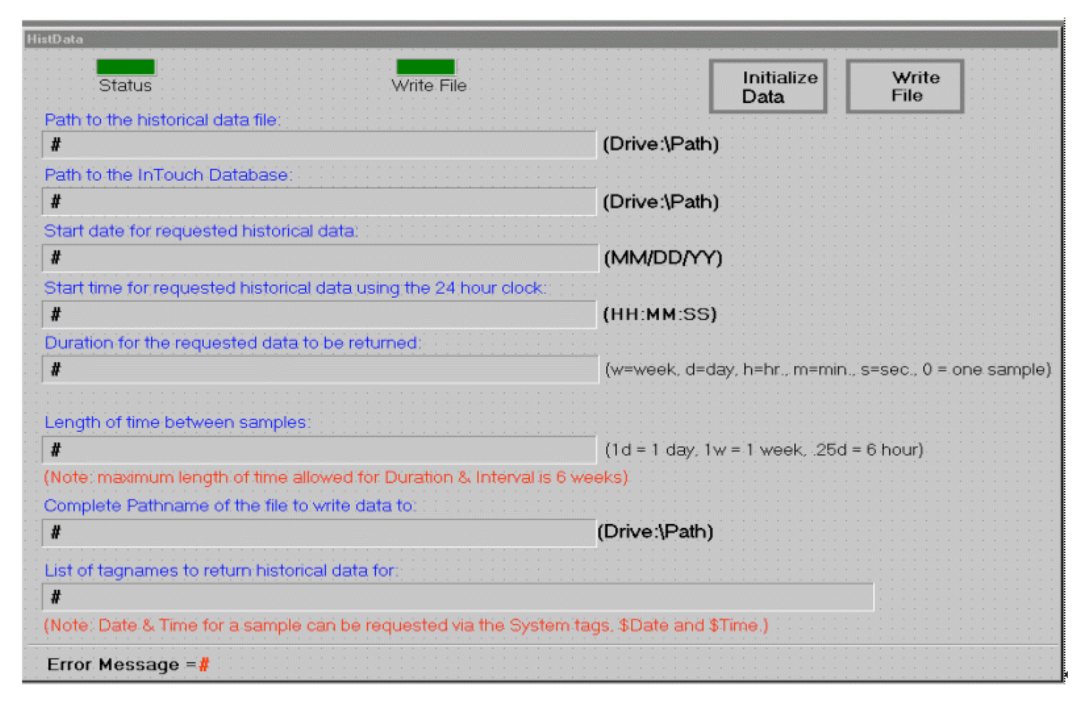
标记名	I/O 标记类型	访问名	项目
HDWSendData	离散	Viewstream1	**SendData**
HDWData	消息	Viewstream1	**Data**

备注 如果您在创建并使用这些标记名后请求大量的信息，则会收到错误消息：请求的数据太多 - 请缩短持续时间或减少标记数。

使用 HistData 向导时，会自动创建 HistData 标记名和 “访问名”。要使用 HistData 向导：

1. 单击 Wizard 工具。
2. 打开向导的 “趋势”组。
3. 双击 HistData 向导并将其粘贴到窗口中。
4. 双击 HistData 向导并给它指定一个 “历史趋势”型标记名。
5. 单击**确定**以创建 HistData 标记名。

在创建 I/O 型标记名之后，创建一个名称为 **HistData** 的新窗口，如下例所示：



号链接至一个输入链接。例如， # 号与 **HDWDataDir** 标记名有一个 “用户输入 / 字符串”链接。用户输入链接可供您在运行时更改标记名的值。

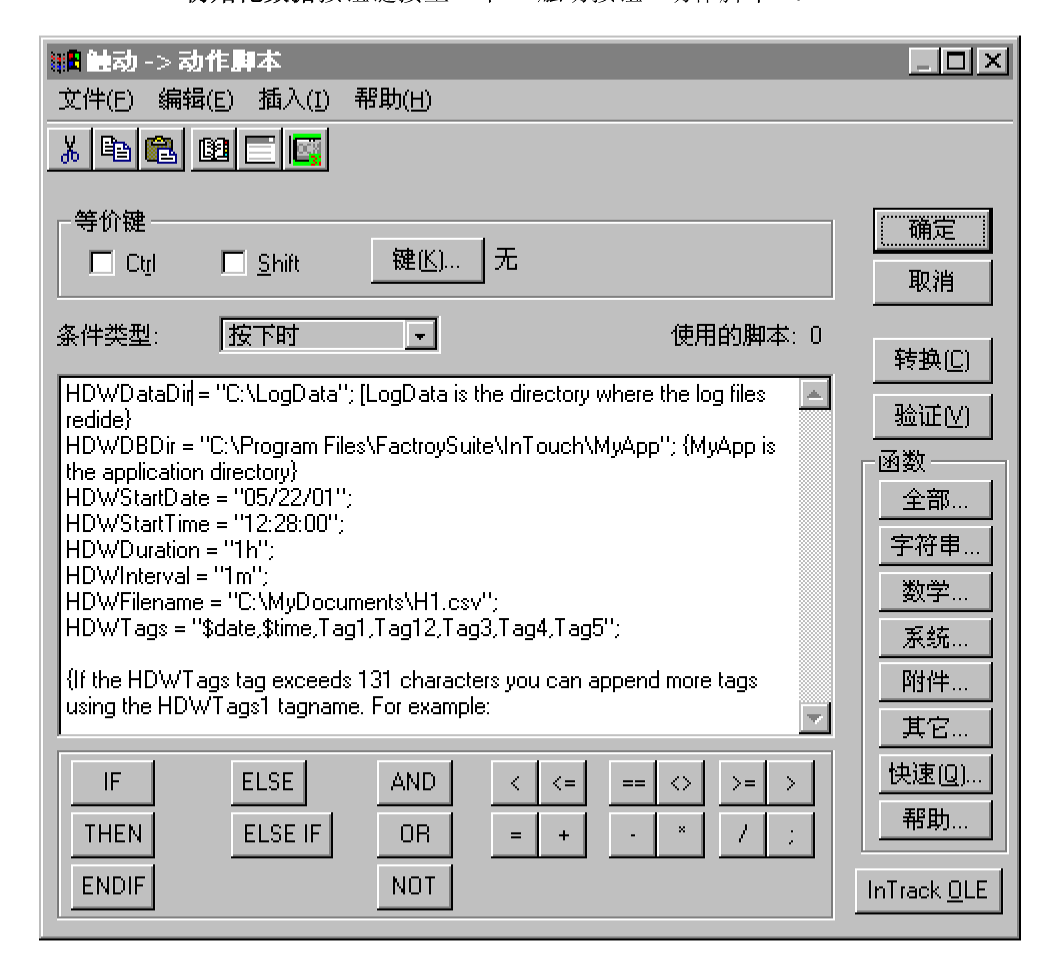
Status 按钮链接至基于标记名 **HDWStatus** 的填充色 - 离散表达式：



写入文件按钮链接至基于标记名 **HDWWriteFile** 的填充颜色 - 离散表达式：



初始化数据按钮链接至一个 “触动按钮 - 动作脚本”。



按下**初始化数据**按钮时， HistData 项目即被初始化为所需的值。如有必要，也可以通过使用 “用户输入链接”在运行时更改这些值。

写入文件按钮链接至 “触动按钮 - 动作脚本”：



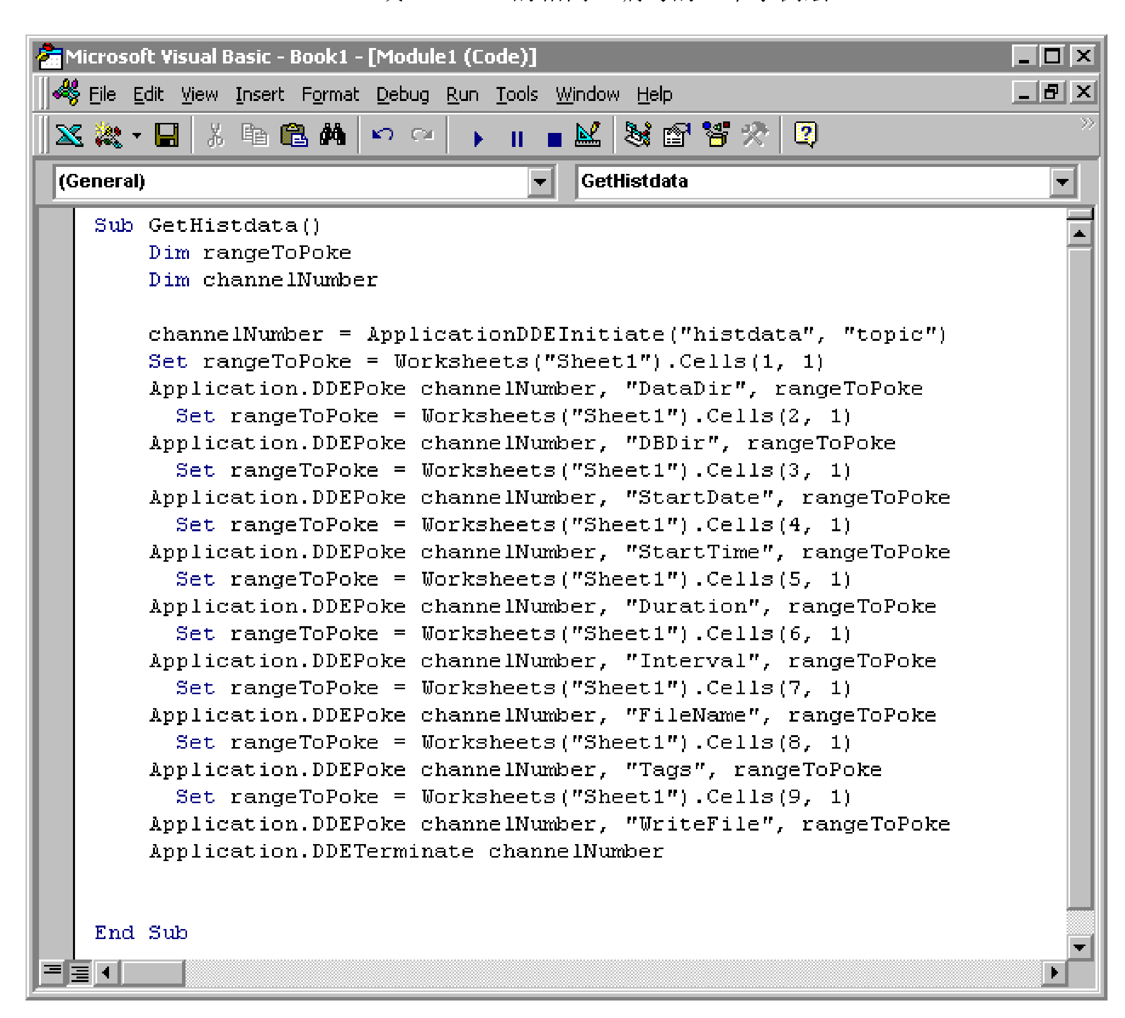
单击 **WriteFile** 按钮时，可生成 .CSV 文件。

在配置并保存上述所有项目之后，启动 **HistData** 并将其最小化。下一步，启动 WindowViewer 并打开 HistData 窗口。单击 **Initialize** 按钮，并根据需要修改 HistData 项目，然后单击 **WriteFile** 按钮。如果操作成功，则 **Status** （状态）值将变为 **ON** （打开），并显示与 **ON** （打开）状态关联的颜色。如果操作不成功，则 **Status** 的值将变为 **OFF** （关闭），并显示**错误消息**来指明出错原因。

在 **Excel** 中使用 **HistData**

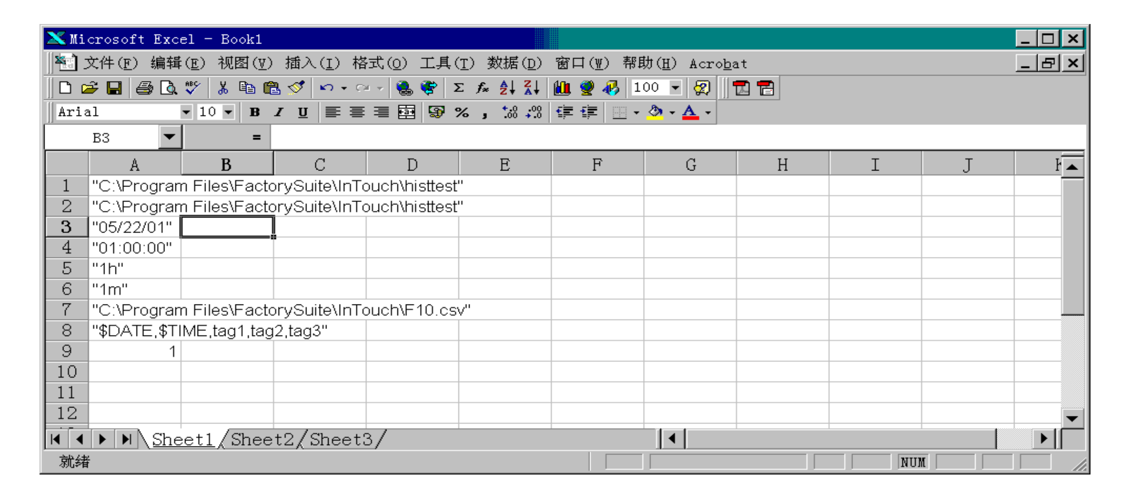
HistData 程序会对 Microsoft Excel 等产品的 **INITIATE**、 **POKE** 和 **TERMINATE** 函数作出响应。带**关键字** （内部数据库项目）的 **POKE** 函数用于设置定义查询的参数。正确设置查询之后，程序会运行宏来请求所选的历史数据。

您可以编写 Excel 宏与 HistData 交互操作。下面是用 Excel 5.0VBA （与 Excel 7.0 或 Excel 97 的相同）编写的一个示例宏：



插入的数据包含于 Excel 电子表格 spreadsheet1 中名称为 Sheet1 的单元格。

下面是 Sheet1 的一个示例，它包含要插入的数据：



常见的 HistData 错误消息

错误消息	原因 / 解决方案
请求的数据太多 - 请缩短持续时间或减少标记名数。	使用 SendData 项目并请求过多的数据时，便会出现此错误。如果唯一的目的便是创建一个 .CSV 文件，用它储存来自加密日志文件的数据，则*请不要使用* ***SendData***。
无法打开文件 **C:\FILES1\HISTDATA.CSV**	目录 **Files1** 不存在。此外，还请检查路径的拼写是否正确。
无法打开文件 **C:\FILES**	未定义 CSV 文件。
DATADIR 项目无效	给 DataDir 项目指定的目录不存在。请检查拼写是否正确。
STARTDATE 项目无效	发送至 StartDate 项目的格式无效。
未找到日志文件	给 DataDir 项目的指定路径中没有请求日期的日志文件。
在数据库中找不到标记名 **TAGX**	标记名字典中不存在所请求的标记名（在本例中是 **TAGX**）。请检查标记名的拼写是否正确。
在下列目录中找不到 **tagname.x**：**C:\IT6.0B\HISTEST**	文件 **tagname.x** 不存在或已损坏。

第 13 章

I/O 通讯

InTouch 使用 Microsoft “动态数据交换”（Dynamic Data Exchange，简称 DDE）、FastDDE、NetDDE 以及 Wonderware SuiteLink 协议与其它的 Windows 程序、 Wonderware I/O 服务器以及同现实世界进行通讯的第三方 “I/O 服务器”程序进行通讯。

目录

支持的通讯协议

“动态数据交换” (DDE) 是 Microsoft 开发的一种通讯协议，可供 Windows 环境下的应用程序彼此发送 / 接收数据与指令。它在同时运行的两个应用程序之间实现一种客户端与服务器关系。*服务器*应用程序提供数据，并接受对这些数据感兴趣的任何其它应用程序的请求。发出请求的应用程序被称作*客户端*。有些应用程序 （如 InTouch 与 Microsoft Excel）可以同时作为*客户端*与*服务器*。

FastDDE 提供了一种方法，可以将许多专用的 Wonderware DDE 消息压缩成单个 Microsoft DDE 消息。由于减少了*客户端*与*服务器*之间所需的 DDE 事务总数，此种压缩处理可提高效率与性能。尽管 Wonderware 的 FastDDE 已将 DDE 应用扩展到我们所在的行业，但是在分布式环境中，此种扩展正被推向其性能极限。

NetDDE 扩展了标准的 Windows DDE 功能，将局域网上及通过串行端口的通讯也囊括在内。各种可用的网络扩展可供在不同计算机上运行的应用程序通过网络或调制解调器建立 DDE 链接。例如， NetDDE 支持在 IBM PC 上运行的应用程序通过局域网或调制解调器同 VMS 和 UNIX 等操作环境下的非 PC 平台上运行且支持 DDE 的应用程序之间进行动态数据交换。

Wonderware SuiteLink 使用基于 TCP/IP 的协议。 SuiteLink 专门设计用于满足象数据完整性、高吞吐量以及方便诊断这样的工业需要。 Microsoft Windows NT 与 Windows 2000 均支持此种通讯标准。

SuiteLink 并非 DDE、 FastDDE 或 NetDDE 的替代品。客户端与服务器之间的每个连接均依赖于网络状况。 SuiteLink 是专为高速工业应用而设计的，它具有下列功能：

“数值时间质量”（Value Time Quality，简称 VTQ）在传输给支持 VTQ 的客户端的所有数据值上均插入一个时间标签与质量指示器。

通过使用 Microsoft Windows NT 操作系统的性能监视器，可以对数据吞吐量、服务器负载、计算机资源消耗及网络传输等进行非常全面的诊断。此项功能对于规划和维护分布式工业网络而言至关重要。

不管应用程序是位于单个节点还是分布于多个节点，均可在应用程序之间可维持一致的高数据容量。

网络传输协议是使用 Microsoft 标准 Winsock 接口的 TCP/IP。

Wonderware SuiteLink

Wonderware SuiteLink 使用基于 TCP/IP 的协议。 SuiteLink 专门设计用于满足象数据完整性、高吞吐量以及方便诊断这样的工业需要。 Microsoft Windows NT 和 Windows 2000 均支持此种协议标准。

要使用 SuiteLink 通讯协议

1. Microsoft TCP/IP 必须已配置并能正常工作。
2. 必须使用不超过 15 个字符的计算机名 （**节点名**）。

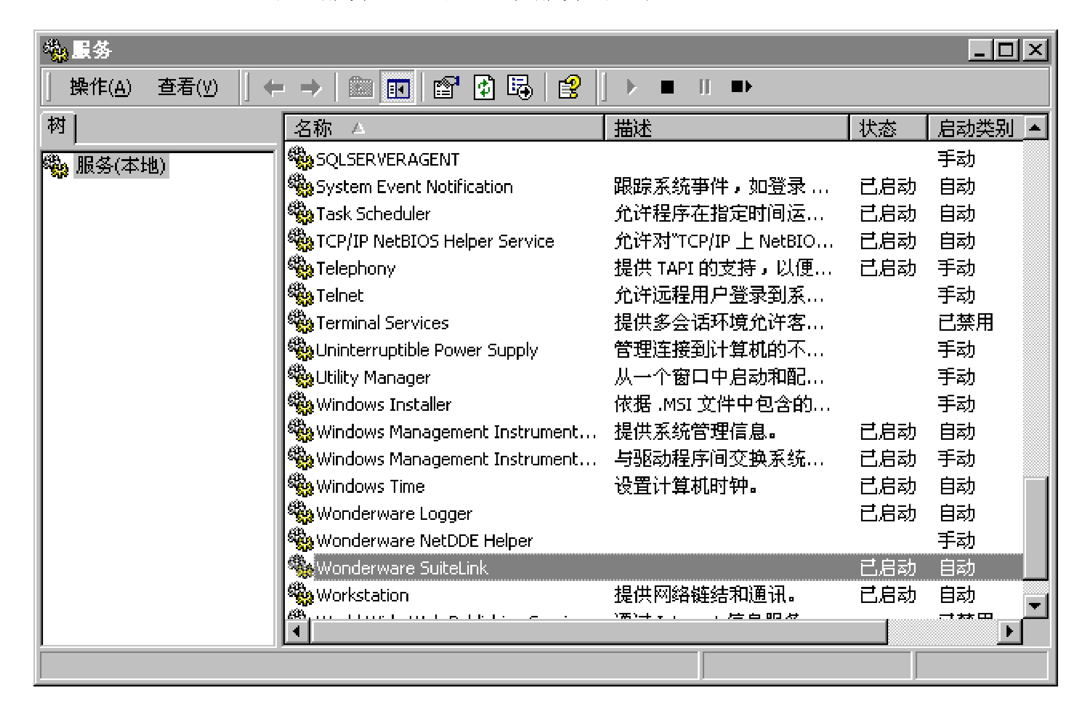
   如需有关安装与配置 Microsoft TCP/IP 的详细信息，请参阅 Microsoft Windows 操作系统文档。
3. Wonderware SuiteLink 必须作为服务运行。如果由于某种原因导致 SuiteLink 已经停止，则您需要再次启动它。（安装 InTouch 时会自动安装 SuiteLink，并且在缺省情况下，它会被配置成自动启动的 “NT 服务”）。

   如需有关 Windows NT 服务的详细信息，请参阅附录 A “InTouch Windows NT 服务综述”。

要将 SuiteLink 配置成作为 NT 服务启动

1. 打开 Windows “控制面板”。

2. 双击**服务**。此时会出现**服务**对话框。



3. 选择 **Wonderware SuiteLink**，然后单击**启动**。

4. 单击**关闭**。

InTouch I/O 地址命名惯例

InTouch 使用三个部分组成的命名惯例来识别 “I/O 服务器”程序中的数据元素，这三个部分是*应用程序名*、*主题名*以及*项目名*。要从另一个应用程序获取数据，*客户端*程序 (InTouch) 需要通过指定上述三个项目来打开通往*服务器*程序的通道。

为了让 InTouch 从其它应用程序获取数据值，它还必须了解提供数据值的*应用程序*的名称、应用程序中包含数据值的*主题*的名称，以及该*主题*中特定*项目*的名称。此外， InTouch 还需要了解数据的类型：离散型、整型、实型 （浮点数），还是消息型 （字符串）。此信息确定了在 InTouch 数据库中定义标记名时它的 I/O 类型。现在，当 WindowViewer 运行时，它会自动执行获取与保持此*项目*的值所需的全部动作。

例如，在 Excel 中，*应用程序名*是 "Excel"，*主题名*是包含数据的特定工作表的名称，*项目名*是工作表上要从中读取数据或向其中写入数据的单元格的标识。

InTouch I/O 地址

另一个 Windows 应用程序从 InTouch 请求提供数据值时，它也必须了解这三个 I/O 地址项。下面介绍 InTouch 的 I/O 地址命名惯例：

VIEW *（应用程序名）*确定包含数据元素的 InTouch 运行时程序。

TAGNAME *（主题名）*是读取 / 写入 InTouch 数据库中的标记名时总是会用到的词。

ActualTagname *（项目名）*是 "InTouch 标记名字典" 中为该项目定义的实际标记名。

例如，要从 Excel 中访问 InTouch 中的一个数据值，则需要将在其中写入数据值的单元格中输入 "DDE 远程引用" 公式：

```
VIEW|TAGNAME!'ActualTagname'
```

备注 如果您联网使用 Wonderware NetDDE，则 I/O 地址的*应用程序名*部分必须以远程节点的名称作前缀，并且要在前面加两个反斜杠，后面再跟一个反斜杠。例如：

```
\\NodeName\VIEW|TAGNAME!'ActualTagname'
```

InTouch 访问名

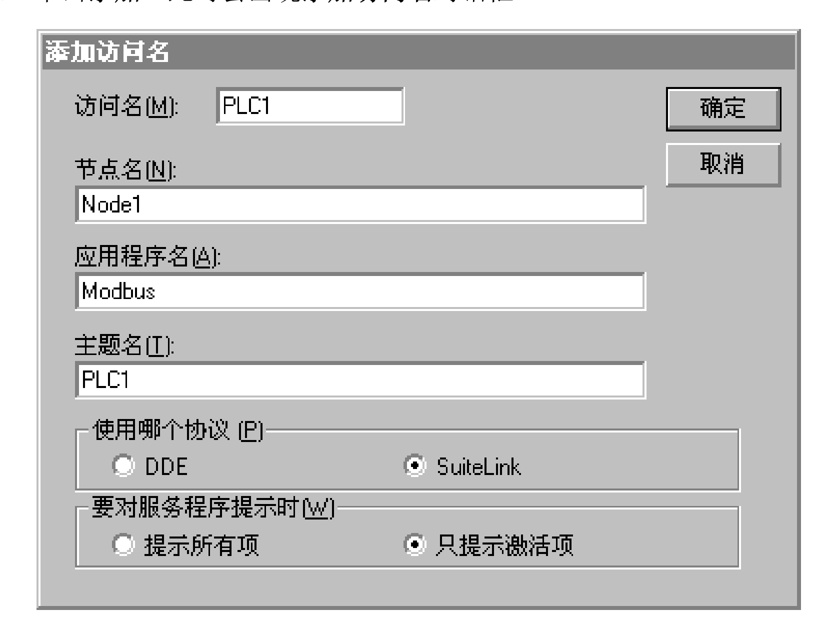
创建 I/O 型标记名或远程标记名引用时，必须将它们与 "访问名" 关联起来。 "访问名" 包含用于同其它 I/O 数据源进行通讯的信息，包括节点名、应用程序名及主题名。

要创建访问名

1. 在**特别**菜单上，单击**访问名**，或者在 "应用程序浏览器" 中的**配置**下，双击**访问名**。此时会出现**访问名**对话框。
2. 在 "应用程序浏览器" 中，你可以右击**访问名**，然后单击**打开**。您也可以在 "标记名字典" 中定义 I/O 型标记名时创建 "访问名"。



3. 单击**添加**。此时会出现**添加访问名**对话框。



4. 在**访问名**框中，输入您希望 InTouch 使用此 “访问名”时所用的名称。（为简单起见，在此请使用与*主题名*相同的名称）。

   InTouch 使用 “访问名”引用实时 I/O 数据。每个 “访问名”均相当于一个 I/O 地址，可包含 “节点”、“应用程序”及 “主题”。在分布式应用程序中，既可以将 I/O 引用设置为指向网络 “I/O 服务器”的全局地址，也可以将它设置为指向本地 “I/O 服务器”的本地地址。

5. 如果数据位于网络 “I/O 服务器”中，请在**节点名**框中输入远程节点的名称。

6. 在**应用程序名**框中，输入将从中获取数据值的 “I/O 服务器”程序实际的程序名。在本例中，数值来自 Wonderware Modbus I/O 服务器，因此使用 **MODBUS**。切勿输入程序名的 **.exe** 扩展名部分。

7. 在**主题名**框中，输入您要访问的*主题名*。 **主题名**是特定于应用程序的数据元素子组。在数据来自 Wonderware “I/O 服务器”程序的情况下，*主题名*与在 “I/O 服务器”程序中为*主题*配置的名称完全相同。与 Microsoft Excel 进行通讯时，*主题名*必须是保存工作表时指定的名称。例如 Book1.xls。

8. 选择您要使用的协议。

   如需有关详细信息，请参阅 “支持的通讯协议”。

9. 选择您要用于提示服务器的选项：

选项	定义
提示所有项	轮询所有数据，无论它是否在可见的窗口中，或是否已报警、记录、绘制趋势或用在脚本中。选择此选项将影响性能，所以不建议您使用。

只提示激活项	只轮询可见的窗口，以及已报警、记录或用在任何脚本中的数据点。 **备注**除非出现在可见的窗口中，否则触动按钮动作脚本不会被轮询。

10. 单击**确定**以接受新 “访问名” 并关闭对话框。此时**访问名**对话框会重新出现，并显示在列表中处于选定状态的新 “访问名”。



11. 单击**关闭**，以关闭对话框并返回到标记名定义。

要修改或删除访问名

1. 在**特别**菜单上，单击**访问名**，或者在 “应用程序浏览器” 中的**配置**下，双击**访问名**。此时会出现**访问名**对话框。
2. 在 “应用程序浏览器” 中，你可以右击**访问名**，然后单击**打开**。



3. 要更改 “访问名” 的定义，请从列表中选择它，然后单击**修改**。此时会出现**修改访问名**对话框。进行所需的更改，然后单击**确定**。此时会再次出现**访问名**对话框。单击**关闭**如果您需要修改其它已定义的 “访问名”，请重复此过程。
4. 要删除 “访问名”，请从列表中选择它，然后单击**删除**。此时会出现一个消息框，要求您确认删除选定的 “访问名”。单击**是**以删除它，或单击**否**以取消删除。单击**关闭**如果您需要删除其它已定义的 “访问名”，请重复此过程。

备注 与标记名关联的 “访问名”无法删除。

在 InTouch 中定义 I/O 项目

在 “标记名字典”中定义 I/O 型标记名时， InTouch 可以接收来自其它本地或远程 Windows 应用程序的数据。每个 I/O 型标记名都引用 “I/O 服务器”程序中的一个有效*项目*。

如需有关分布式应用程序的详细信息，请参阅第 5 章 “建立分布式应用程序”。

要定义一个 I/O 型标记名

1. 在**特别**菜单上，单击**标记名字典**，或者在 “应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。
2. 单击**新建**。此时**标记名**框会被清除。

提示 如果您右击 “标记名字典”对话框中的任意文本输入框，则会出现一个菜单，显示可应用于所选文本的各条命令。



第一次访问 “标记名字典”时，会显示内部系统标记名 **$AccessLevel** 的定义。在 “标记名字典”中定义标记名之后，当您再次访问它时，则会显示最近编辑的标记名的定义。

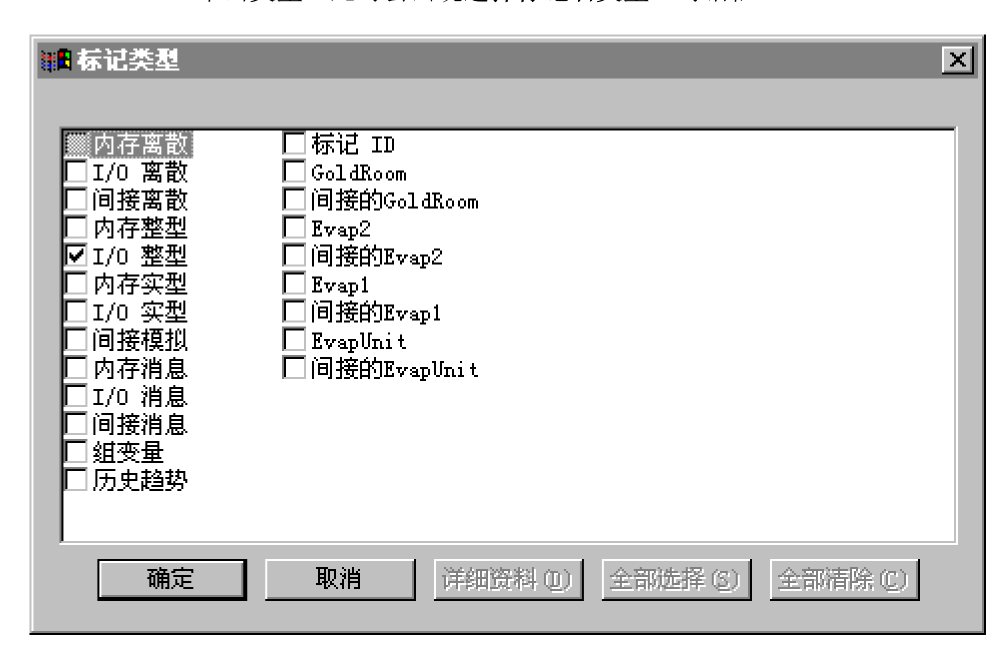
3. 在**标记名**框中，输入要给新标记名使用的名称。

   标记名最长可达 32 个字符，但必须以希腊字符 （A-Z 或 a-z）开头。其余字符可以是 A-Z、a-z、0-9、!、@、-、?、#、$、%、_、\ 以及 &。

   标记名也会自动编号。例如，如果您输入并保存标记名 R4001，然后单击**新建**，则新标记名会自动编号为 R4002。如果标记名包含由一个字符分隔的数字，则会根据 InTouch 找到的第一个整数给它自动编号。例如， N7-0 将编号为 N7-1。注意这里只允许发生正向变化。例如， R4002 到 R4003、 R4003 到 R4004，依此类推。

   标记名不能使用单词 **RetVal**。这是一个保留字。如果您试图使用这个字，然后尝试编辑 QuickFunction，则您会收到一则错误消息 “变量不得使用该名称。标记已存在”。

4. 单击**类型**。此时会出现**选择标记名类型**”对话框。



5. 根据下表给标记名选择 I/O 类型：

标记类型	**输入 / 输出值**
I/O 离散	True、 On、 Yes (1) 或 False、 Off、 No (0)
I/O 整型	整数
I/O 实型	浮点小数
I/O 消息	字符串

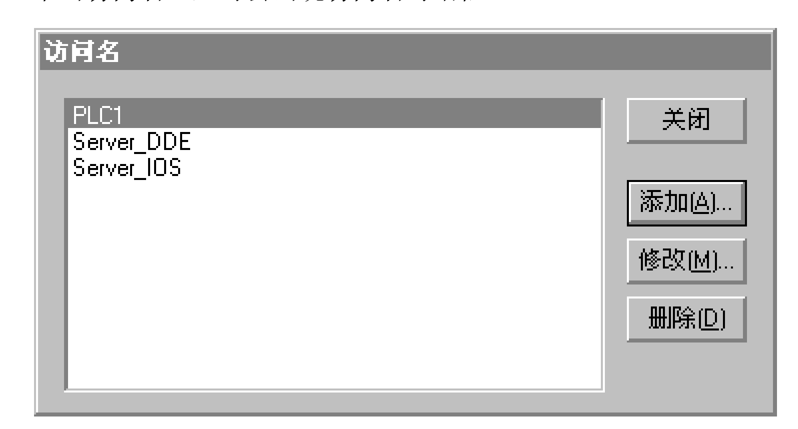
6. 在选择 I/O 类型之后，单击**确定**。此时会出现所选的 I/O 类型对应的“详细”对话框。例如，如果您选择 “I/O 整型”，则会出现下面的对话框：



提示 如果 “详细”对话框没有出现，请单击屏幕顶部的**详细**。

7. 指定定义*项目*所需的全部详细资料。

8. 单击**访问名**。此时会出现**访问名**对话框。



9. 双击列表中您要使用的 “访问名”，或者选择它，然后单击**关闭**。

   此时所选的的 “访问名”（现在与此标记名定义相关联）出现在 “详细”对话框中**访问名**按钮旁边。例如：

初始值(V): 0　最小工程单位(N): -99　最大工程单位(X): 99
死区(D): 0　最小原始数据(W): -99　最大原始数据(A): 9
工程单位(E):　记录死区(B): 0　转换　线性(L)　平方根(S)
访问名(M): ...　PLC1
项目(I):　使用标记名作为项目名(U)

10. 在**项目**框中，输入 “I/O 服务器”程序中数据值的*项目名*。

备注 “标记名”是用在 InTouch 中引用数据值的名称，了解这点非常重要。**项目**则是远程 Windows 应用程序引用相同值时使用的名称。这些名称不必相同，但如有可能，建议您使用相同的名称。此外，如果**项目**是 Excel 中的一个单元格，则必须将它指定为在 Excel 中给它的定义名，或指定为它的行 / 列标识。例如， R1C1。

---

11. 单击**关闭**。

    如需有关定义 I/O 标记名的详细信息，请参阅第 6 章 “标记名字典”。

监视 I/O 对话的状态

WindowViewer 支持内置的*主题名* **IOStatus** （在 InTouch 7.0 以前版本中称为 **DDEStatus**），它可以用于监视特定 I/O 对话的状态。

使用 IOStatus 主题名

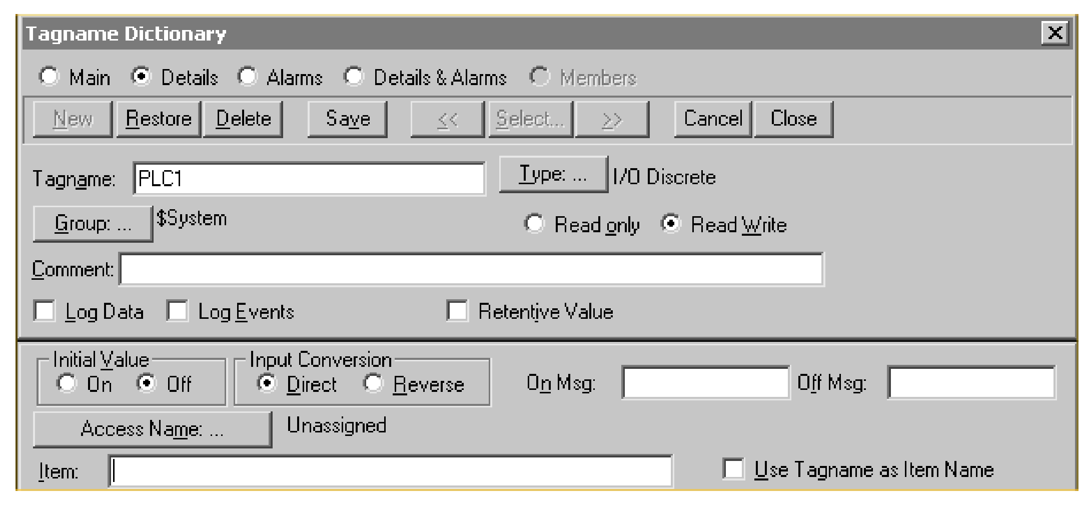
我们假设 WindowViewer (View) 正在与一个连接到 PLC 的“Wonderware 仿真 I/O 服务器”通讯，该 PLC 在 “I/O 服务器”的*主题名*定义为 **PLC1**。

（“仿真”是一个通用的 “Wonderware I/O 服务器”，通常用作培训工具。 FactorySuite 中便包含有 “仿真”程序）。

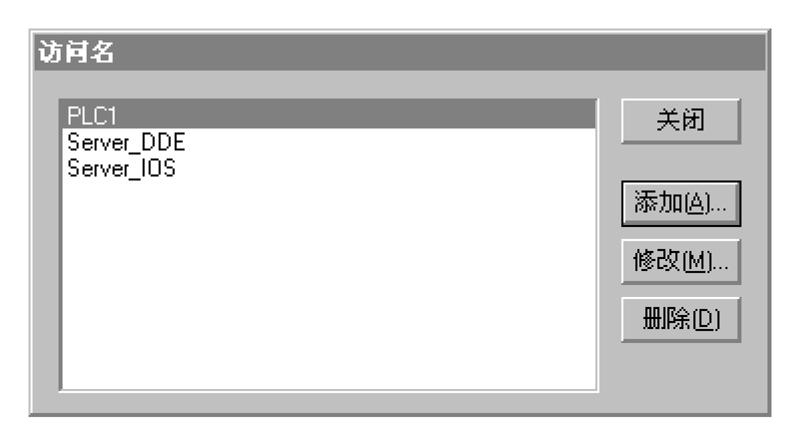
要监视 I/O 通讯的状态

1. 在**特别**菜单上，单击**标记名字典**，或者在 “应用程序浏览器”中，双击**标记名字典**。此时会出现**标记名字典**对话框。
2. 创建一个 **I/O 离散**型标记名。（在本例中，为简单起见，我们将标记名设置为与要监视的*主题名*相同）。

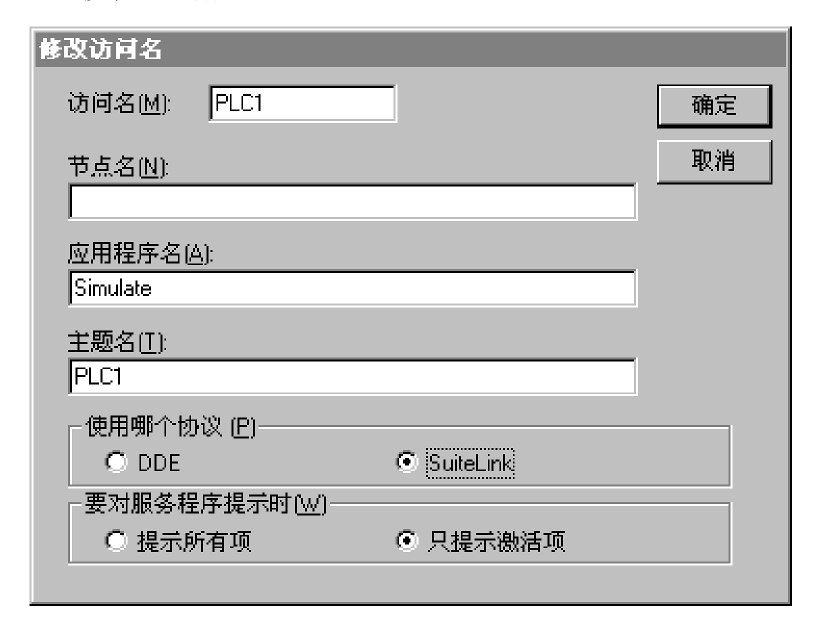
   在使用 **IOStatus** 监视 I/O 对话时，您至少要为所监视的 “访问名”定义一个 I/O 型标记名。



3. 单击**访问名**，将标记名指定给一个 “访问名”定义，它定义其*主题名*的 **IOStatus**。此时会出现**访问名**对话框。



请注意，目前已存在一个名称为 **PLC1** （我们要监视的主题）的“访问名”定义。要确保 “访问名”正确 （其**主题名**为 **PLC1**），请单击**修改**以查看其定义。



在本例中，因为我们将标记名定义为与**主题名**相同，因此可以很轻松地找到包含正确*主题名*的 “访问名”。

4. 单击**取消**关闭对话框，并返回到最初的**访问名定义**对话框。
5. 单击**添加**。此时会出现**添加访问名**对话框。

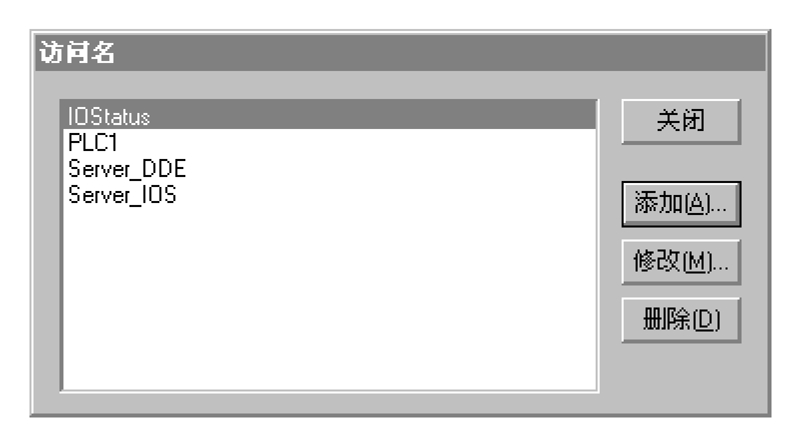


6. 在**访问名**框中，输入 **IOStatus**。

   因为要在 WindowViewer 中监视状态，所以请在**应用程序名**框中，输入 "View"。

7. 在**主题名**框中，输入 InTouch 内部*主题* **IOStatus**。

8. 选择**只提示激活项**。
9. 单击**确定**以关闭对话框。此时会再次出现最初的**访问名定义**对话框，在列表中显示新的**访问名 IOStatus**：



10. 单击**关闭**以关闭对话框，并将这个新的**访问名**与 **I/O 离散**型标记名关联起来：



11. 在**项目**框中，输入要监视的实际**主题名**的**访问名**。在本例中是 **PLC1**。

    因为标记名与**主题名**相同，您可以选择**使用标记名作为项目名**，使之自动输入到**项目**框。

    备注　使用内置的主题 **IOStatus**（在 InTouch 7.0 以前版本中称为 **DDEStatus**）监视 I/O 对话时，**项目**框总是使用您在**访问名**框中输入的名称。

在 Excel 中使用 IOStatus 主题名

通过在工作表单元格的公式中输入相同的信息，您也可以使用 Excel 执行此种类型的监视活动。例如，要监视上述主题，请输入下面的公式：

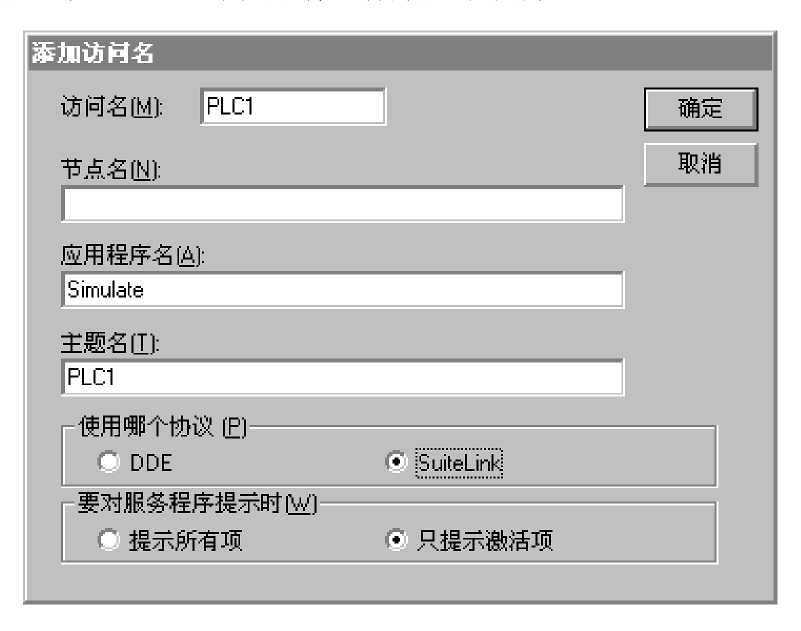
```
=view|IOStatus!'PLC1'
```

监视 I/O 服务器通讯状态

对于使用的每个*主题名*，都有一个内置的离散型*项目* **Status**，它可以用于监视与“I/O 服务器”程序通讯的状态。与设备通讯失败时（如电缆断开，PLC 关机等），**Status** 会被设置为 "0"；通讯成功时，它会被设置为 "1"。

备注 使用 "Status" 项目监视主题的状态时，所监视的主题至少要有一个活动的 I/O 点。

在 InTouch 中，通过定义一个标记名，然后通过将 **Status** 用作*项目名*来配置一个*主题*，并将此标记名与主题关联起来，这样就可以读取服务器通讯的状态。例如，如果 WindowViewer 使用 "Wonderware 仿真 I/O 服务器" 与 PLC 通讯，则可以将 "访问名" 定义为：



要监视主题 *PLC1* 的所有通讯状态，您需要创建下列标记名定义：



如需有关排解 I/O 通讯疑难的详细信息，请参阅《Wonderware I/O 服务器用户指南》。

提示 在 Excel 中，您可以通过在单元格中输入下列公式来读取 PLC 通讯的状态：

=SIMULATE|PLC1!'STATUS'

监视多个输入设备的状态

本节介绍如何使用多个输入设备来显示某个对象的状态。

示例 1

在本例中，我们使用两个输入来监视弹簧复位机动阀的状态。两个输入代表安装在阀体上的一对限位开关。其中一个输入只有在阀处于打开位置时才接通，而在阀移动或关闭时断开。另一个输入则是在阀处于关闭位置时接通，而在阀移动或处于打开位置时断开。这两个输入的真值表如下所示：

真值表 1

输入#1 (打开)	输入#2 (关闭)	阀位置	结果
1	0	打开	1 + 0 = **1**
0	1	关闭	0 + 1 = **1**
0	0	InTravel	0 + 0 = **0**
1	1	InValid Position	1 + 1 = **1**

0 = OFF 1 = ON

通过将关闭的输入乘以 2，可以对这两个输入进行加权平均，因此阀位置的结果变为如下内容：

真值表 2
(输入#2x2)

输入#1 (打开)	输入#2 (关闭)	阀位置	结果
1	0 x 2 = 0	打开	1 + 0 = **1**
0	1 x 2 = 2	关闭	0 + 2 = **2**
0	0 x 2 = 0	InTravel	0 + 0 = **0**
1	1 x 2 = 2	InValid Position	1 + 2 = **3**

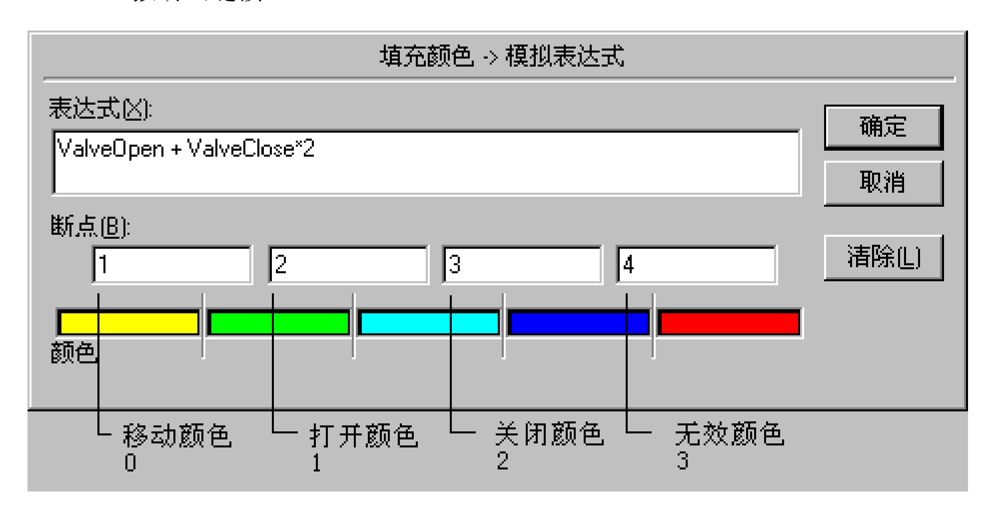
0 = OFF 1 = ON

备注 无效位置可用于显示损坏的限位开关。

现在，由于阀位置之间存在显著的数值差异，因此我们可以使用一个**填充颜色 - 模拟**动画链接来显示阀的状态。

如需有关创建动画链接的详细信息，请参阅第 7 章 “创建动画链接”。

现在已创建了两个 I/O 离散型标记名。一个用于阀打开输入，一个用于阀关闭输入。例如， **ValveOpen** 和 **ValveClosed**。然后，我们创建一个对象来显示阀的状态。此对象通过使用下列属性指定给一个**填充颜色 - 模拟**动画链接：



示例 **2**

在本例中，我们在现有的两个输入的基础上再增加一个输入。这个新的输入是导致阀打开的实际输出。新的输入将在阀打开或正在打开时接通，而在阀关闭或正在关闭时断开。此时，真值表如下所示：

真值表 3

输入#1 (打开)	输入#2 (关闭)	输入#3 (打开)	阀位置	结果
0	0	1	正在打开	0 + 0 + 1 = **1**
1	0	1	打开	1 + 0 + 1 = **2**
0	0	0	正在关闭	0 + 0 + 0 = **0**
0	1	0	关闭	0 + 1 + 0 = **1**

0 = OFF 1 = ON

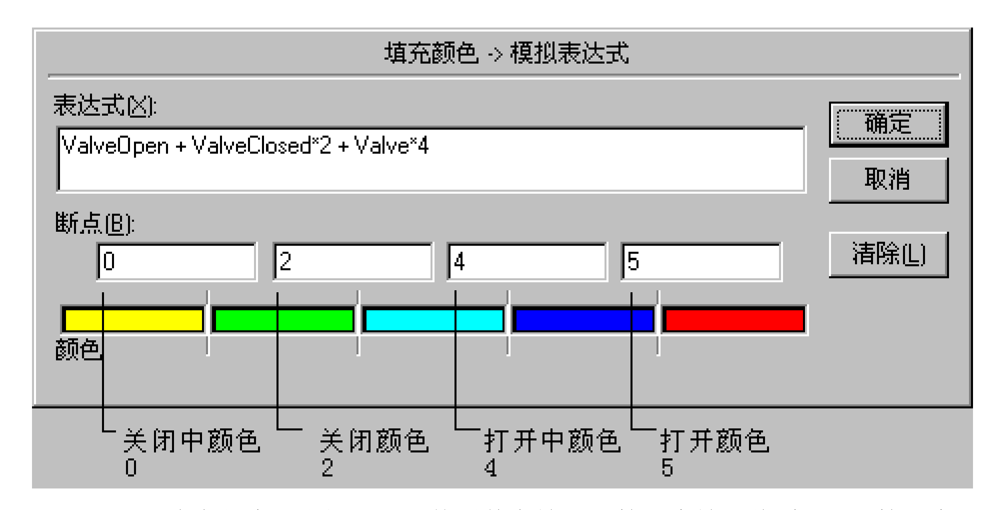
我们再次对输入进行加权平均。如前所述，我们将关闭输入乘以 2，将新输入乘以 4，得出下面的结果：

真值表 4
(输入 #2x2 输入 #3x4)

输入#1 (打开)	输入#2 (关闭)	输入#3 (打开)	阀位置	结果
0	0 x 2 = 0	1 x 4 = 4	正在打开	0 + 0 + 4 = **4**
1	0 x 2 = 0	1 x 4 = 4	打开	1 + 0 + 4 = **5**
0	0 x 2 = 0	0 x 4 = 0	正在关闭	0 + 0 + 0 = **0**
0	1 x 2 = 2	0 x 4 = 0	关闭	0 + 2 + 0 = **2**

0 = OFF 1 = ON

我们再为新的打开输入创建一个 “I/O 离散型”标记名 (**Valve**)，并使用下列属性将它指定给一个**填充颜色 - 模拟**动画链接：



通过此方法，我们还可以使用其它输入。第 4 个输入将乘以 8，第 5 个输入将乘以 16，依此类推。

第 14 章

Terminal Services for InTouch

本章综述 “终端服务”的概念及其功能和优点。它假设您已具备安装和使用 “终端服务”的知识。如需有关详细信息，请参阅 *Terminal Services for InTouch 部署指南*。

目录

简介

“终端服务”是一种可配置的服务，它包含在 Microsoft® Windows® 2000 Server 操作系统中，可供您从服务器集中运行 32 位基于 Windows 的应用程序。“终端服务”已完全集成到 Windows 2000 Server 内核。许多不同的桌面平台 （MS-DOS®、 Windows、 Macintosh、 UNIX、 Java 及其它）均有 “终端服务”仿真程序客户端。非基于 Windows 的平台则需要第三方附加软件，如 Citrix® MetaFrame。

与传统型客户端 / 服务器环境不同，在 Windows 2000 Server 中启用 “终端服务”时，所有的应用程序处理均在服务器上进行。“终端服务”的客户端不在本地执行任何应用程序处理，而只是显示应用程序的输出。“终端服务”技术只向客户端传输应用程序的外观显示 — 图形用户界面（简称 GUI）。每个用户登录后，仅感觉到自己的会话，这是由服务器操作系统透明管理的，它独立于其它所有的客户端会话。

实施 “终端服务” 多用户、以服务器为核心的计算环境的最大优点是，系统管理及桌面 / 系统部署得到了简化。通过实施以服务器为核心的计算环境，您可以显著降低总拥有成本以及运行成本。

从应用程序开发的角度来看，“终端服务” 的最大优点之一是，那些性能优良的 16 位或 32 位基于 Windows 的应用程序可以照旧运行 — 在 “终端服务” 下运行它们不需要进行任何程序设计方面的修改。布谷，这并不表示您现有的全部应用程序在 “终端服务” 下也能很好地运行。理解如何设计应用程序以充分利用 Windows 2000 “终端服务” 的新功能，这点非常重要。

FactorySuite 应用程序在后端服务器上运行应用程序逻辑，同时使用客户端设备进行数据采集或显示，因而比其它应用程序更适于在以服务器为核心的多用户环境下进行部署。

使用 “终端服务” 需要在网络布置方式和考虑用户的方式上进行一些调整。

如需有关详细信息，请参阅 *Terminal Services for InTouch 部署指南。*

备注 Microsoft 和 Citrix 均不建议在 “终端服务器” 上运行 BackOffice 产品，如 SQL Server，除非是用户负载非常小。随着负载的增加，后台进程排队等候运行时，看上去好象是锁定了那样。

采用多用户方法进行计算并非是 “非此即彼” 的决定 ó 多用户计算可极好地与分布式系统或面向客户端 / 服务器的信息构架共存。在许多应用程序环境中，需要采用多种环境使最终用户达到最佳工作效率和最高生产力。

Windows “终端服务” 可安装在任何支持 Windows 2000 Server 的计算机上。它需要约 14 MB 的额外磁盘空间来存放客户端安装文件，但不需要为操作系统准备额外的空间。不过，对于在应用程序服务器模式下使用 Windows “终端服务” 的机器来说，实际要求可能要高些。每个用户都在服务器上执行自己的程序，因此您需要去确定用户到底是如何工作，他们实际的需求是哪些。每套系统都有所不同。

如需有关详细信息，请参阅 “前言” 中的 “系统要求” 部分。

本章旨在向您简要介绍 Windows® 2000 Terminal Services™ 的概念。

终端服务器概念

“终端服务器” 又回到了大型机的计算概念上。所有的应用程序处理和逻辑均在 “终端服务器” 上发生。终端连接到 “终端服务器” 时，它们会打开一个会话并利用它的部分资源。由于 “终端服务器” 提供所有资源，所以客户端的维护和升级工作量得到最大限度的减少。主要注意力又重新转回到中央服务器上来，就像是在大型机中那样。这样可极大简化维护工作。

服务器和客户端之间的通讯是通过 RPD （远程桌面协议）和 ICA （独立计算结构体系）之类的服务器 / 客户端软件来实现的。每个终端需要一个安装好的 RDP 或 ICA 客户端应用程序，将输入数据 （键盘输入、鼠标移动和串行通讯）传输到服务器上的通讯软件。这些信息由服务器进行处理，然后将图形显示 （屏幕图象）返回至终端进行显示。

RDP （远程桌面协议）是一种通讯协议，其服务器端软件包含在 Windows 2000 Server “终端服务”之中。它包括一个实用程序，可以为 16 位和 32 位 Windows 计算机创建 RDP 客户端磁盘，以便通过 TCP/IP 进行通讯。

ICA （独立计算机结构体系）是 Citrix 提供的第三方通讯协议。它与 Citrix Device Services 或 Citrix MetaFrame 服务器端软件一起使用。 ICA 具有比 RDP 更大的灵活性，在 Windows 3.11/95/98/NT 和 Windows 2000 的基础上，增加了适用于 Unix、 Linux、 OS/2、 DOS、 Windows CE、 PDA、 Java 以及 Macintosh 平台的客户端软件。除 TCP/IP 以外， ICA 协议还可以通过 IPX、SPX、NetBIOS 及 RAS 远程访问进行使用。ICA 是 Windows 2000 完全支持的协议。

在传统的计算机上安装 RDP 或 ICA 协议，就可以使之成为一个客户端，在 “终端服务器”上完成处理，并在自己的屏幕上显示 “终端服务器”的桌面。这就是所谓的 “胖客户端”，因为这台计算机有操作系统和硬盘，但是会连接到 “终端服务器”，并在它上面运行应用程序。胖客户端系统常用于延长老化计算机的寿命。通过加载 RDP 或 ICA 客户端并作为终端来运行， Windows 3.11 计算机就可以运行 Windows 2000。胖客户端的缺点是有可能会出现磁盘故障并需要不断维护硬盘。

“瘦客户端”是不带硬盘并设计为同 “终端服务器”一起使用的客户端。这种机器在芯片上内嵌操作系统和客户端软件，或者是从服务器下载操作系统和客户端。它们连接到 “终端服务器”，并在那里运行它们的应用程序。瘦客户端有其优势，因为所需的维护量最少，而且由于消除了发生硬盘故障的机会，因此提高了可靠性。使用网络下载操作系统的瘦客户端可以在不更换芯片或组件的情况下进行系统升级。在服务器上升级下载软件就可升级所有这些瘦客户端。

“总体拥有成本”的理念 （Total Cost of Ownership，简称 TCO）很简单：最大限度增加公司的技术投资回报 (ROI)，同时最大限度减少所投入的成本。您打算减少的成本可以分为两类：*硬成本*和*软成本*。硬成本是采购和配置硬件所需的成本。软成本是与终端用户支持、培训以及生产环境维护有关的成本。采用瘦客户端 （定义为不配硬盘的终端）、终端、新型家电类设备、瘦 PC 机，有希望减少购置计算机硬件，特别是桌面级硬件的*硬成本*，同时减少与系统管理有关的管理性*软成本*。换句话说，通过实施以瘦客户端 / 服务器为核心的计算系统模式， IS 经理可显著降低计算机设备的总拥有成本，同时提高给用户提供的服务水平。

用户环境

通过从 Windows **程序**菜单运行 “终端服务客户端”命令，您可以从客户端访问 “终端服务”。与 “终端服务器”连接之后，客户端机器上的环境看起来与 Windows 2000 Server 和 Windows 2000 Professional 环境一样。由于应用程序不在本机运行，因此对用户而言是完全透明的。

应用程序在客户端桌面上的 “终端服务”窗口中打开。所有应用程序处理均发生在运行 “终端服务”的服务器上，服务器将显示结果发送给所有的客户端。

“终端服务”的多用户系统环境由三个部分组成：

1. **终端服务器**。服务器管理每个客户端会话的计算资源，并为登录上来的每个用户提供独特的环境。服务器接收并处理远程客户端执行的所有键盘输入和鼠标动作，并将操作系统和应用程序的显示输出发送给相应的客户端。
2. **远程桌面协议 (RDP)**。 RDP 支持在客户端和服务器之间进行通讯。 RDP 在将图形界面元素传输给客户端方面进行了优化。 RDP 是应用层协议，它依赖 “传输控制协议 /Internet 协议”（Transmission Control Protocol/Internet Protocol，简称 TCP/IP）在网络上运送信号。 RDP 系以多通道会议的 “国际电讯联盟”（International Telecommunication Union, 简称 ITU）的 T.120 标准为基础。
3. **客户端**。“终端会话”作为本机桌面环境中的一个窗口打开。在该窗口中运行的是 “终端服务器”的远程桌面。计算机或设备只需配备极少量必需的软件，即可建立与服务器的连接并显示用户界面。

终端服务的各项优点

本节介绍使用 “终端服务”的各项功能和优点。

- **更为快速地将 Windows 2000 呈现于桌面**。在从旧桌面移植到 Windows 2000 Professional 时，“终端服务”起到了消除其中差距的桥梁作用，它采用 “虚拟方式”将 Windows 2000 桌面体验带给非计算机类桌面及需要升级硬件才能在本机运行完整的 Windows 2000 操作系统的计算机。在许多不同的桌面平台上，包括 MS-DOS、基于 Windows 的终端、 Macintosh 及 UNIX，均有“终端服务”客户端提供。（与 MS-DOS、 Macintosh 及基于 UNIX 的计算机连接需要用到其它的软件）。
- **充分利用现有的硬件**。“终端服务”允许计算机同时用作瘦客户端和全功能的个人计算机，它扩展了分布式计算模式。计算机可以继续发挥它们在现有网络中的作用，同时还可充当瘦客户端来模拟 Windows 2000 Professional 桌面。
- **远程管理和支持**。“终端服务”为 Windows 2000 Server 提供了远程管理功能，使系统管理员可以从广域网上的任何一台客户端或通过拨号连接来远程管理他们的服务器。管理员可以从某个桌面远程管理 Windows 2000 服务器。管理员可访问系统管理工具，并能够执行所有的管理工作，包括安装软件，就象是在服务器本机上工作一样。
- **取代文字终端**。由于许多 Windows 终端本身就支持在同一台设备上进行终端仿真，所以您可以使用 Windows 终端来取代文字终端。 Windows 终端可让您处理来自大型机系统的数据，以访问更新的 Windows 应用程序，如电子邮件。
- **更为安全可靠**。管理员可利用多层加密技术，根据安全要求以不同的级别 （低、中、高）对 Windows 2000 Server 和 “终端服务”客户端之间传输的所有数据或某些数据进行加密。此外，“终端服务”登录过程包括改变口令、桌面解锁和屏幕保护解锁等项功能。登录过程是加密的，可确保安全传输用户名和口令。“终端服务”支持在服务器和客户端之间采用 40 位和 128 位 （仅适用于美国和加拿大）两种加密方式。

  因为客户端上存储任何应用程序或用户数据，“终端服务”提供了更为有力的安全保障。“终端服务”还支持多层加密机制，可供您在服务器与客户端之间的传输链路有被窃密之可能时使用此项功能。加密级别有三种：低、中、高。所有加密级别均采用 RSA RC4 标准加密技术。

  采用瘦客户端可防止在出现计算机故障的情况下丢失数据因为数据是在 “终端服务器”上处理和存储的，客户端损坏时不会导致数据被破坏。这样就减少了需要加强数据保护的节点数。

  如需有关详细信息，请参阅 《Microsoft Windows® 2000 Server 管理员手册》。

- **远程控制**。管理员可利用 “远程控制” 功能临时控制用户的会话并观察用户的活动。管理员还可以与用户交互，并以其名义执行命令。要充分利用使用 RDP 的 “远程控制” 功能，两个客户端均必须连接到 Windows 2000 “终端服务器”。

备注 在启用会话的远程控制功能之前，您可以通过在客户端上显示信息及要求客户端允许查看或参与会话的方法来通知用户。远程控制的设置可设置为在无需用户同意的情况下进行远程控制。鼠标和键盘输入可以启用，也可关闭。

- **统一部署程序**。通过在 Windows 2000 Server 上运行 “终端服务”，所有的程序执行、数据处理和数据存储均可在服务器上完成，从而实现程序的统一部署。“终端服务” 确保所有客户端均可访问当前版本的程序。软件只需在服务器上安装一次，而不是在整个组织的每台机器上都要安装，从而可以降低单独升级每台计算机所需的相关成本。
- **集中管理**。 Windows “终端服务” 具有集中管理的功能，同时也为特定用户提供了使用 Windows 桌面环境的灵活性。由于 Windows “终端服务” 会话中的所有应用程序只存储在一处 － 服务器上 － 处理能力和系统软件版本的所有升级均可以在服务器级上进行，而不是在桌面级进行，从而免去了耗时的现场升级活动。这种模式下，没有客户端应用软件需要去开发、安装和升级。因此，安装和相关的软件费用会显著降低。用户配置文件可存储在 “终端服务器” 上，因此可以集中管理客户端桌面，从而提高了应用程序的管理效率 （集中备份和存储、版本升级等等）。

  此外， Windows “终端服务” 还允许管理员检查用户会话的进行情况，甚至可以直接控制它。支持人员可以在不离开办公桌的情况下，准确地观察到用户所看到的内容。如果对用户进行了相应地配置，支持人员可以参与会话控制，与用户一起解决难题。因为花在排除故障上的时间更少，维护成本会得以降低，本机感染病毒的危险减小，软件升级更为容易，发生故障的平台也更容易更换，同时不会丢失任何数据。

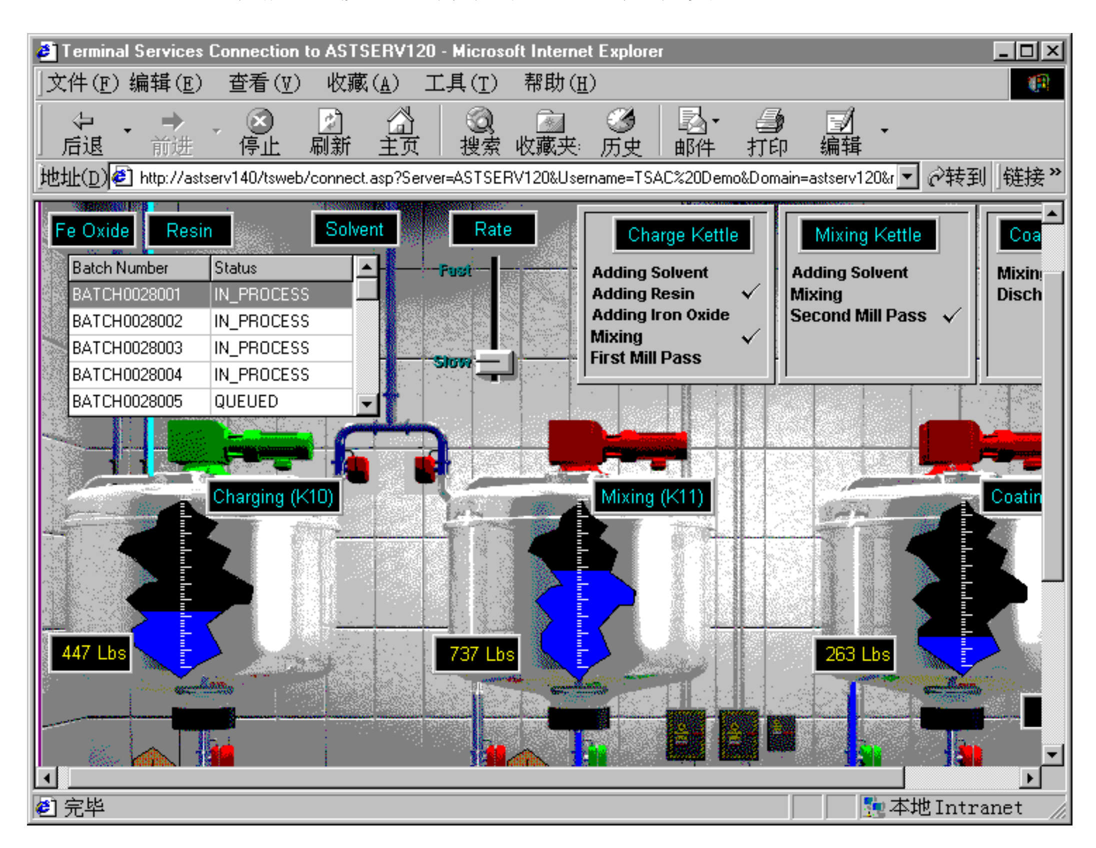
  配置为远程管理模式时， Windows “终端服务” 还可以用作管理工具。管理员可直接从他们的桌面登录到服务器上，进行正常地系统维护而不必坐在服务器控制台旁边。

  通过采用 Windows “终端服务”，您可以将过去在用户桌面去执行的三件工作集中到一起来做：

  1. 应用程序支持和维护
  2. 操作系统维护
  3. 存储管理

终端服务高级客户端

Microsoft 的 “终端服务高级客户端”（Terminal Services Advanced Client，简称 TSAC）是一个基于 Win32Æ 的 ActiveXÆ 控件，可用于在 MicrosoftÆ Internet Explorer 中运行 “终端服务”会话。通过使用 TSAC，现在您可以在 Internet 上运行功能齐备的 InTouch 应用程序，其性能和速度可以与在局域网上运行相媲美。



优点

这个可下载的 ActiveX 控件提供的功能与完整的 “端终服务客户端”几乎一样，其独特之处在于能够通过 Web 实现此项功能。 TSAC 可提供下列优点：

- **在 Internet Explorer 中运行会话**。客户端机器上不必安装终端仿真软件。只需使用 Internet Explorer 4 或更高版本，并在需要时将 URL 地址指向 **终端服务器**即可。
- **快速简便地访问 终端服务器**。 TSAC 特别适合根据需要去访问**终端服务器**。URL 地址可以包含一些可选的字段，例如用户名和屏幕大小，以便访问不同的**终端服务器**，就如同单击 “收藏夹”链接那样简单。

- **公用界面**。与 Internet Explorer 一样的外观使它成为查看 MS Office™ 应用程序、浏览工厂信息、使用 ActiveFactory™ 或 SuiteVoyager™ 进行趋势分析的首选 GUI。

安装

TSAC 是 Microsoft 网站上提供的一个免费 ActiveX 控件。它必须安装在运行 Internet Information Services (IIS) 4.0 或更高版本的计算机上。此依赖性只适用于 Web 服务器。用户可以下载该控件，并可以在所支持的任何 Web 浏览器 （Windows 32 位版的 Internet Explorer 4.x 或 5.x）上查看会话。

备注 如需最新信息，或需要去下载 TSAC，请访问 http://www.microsoft.com/WINDOWS2000/downloads/recommended/TSAC/default.asp

如何使用

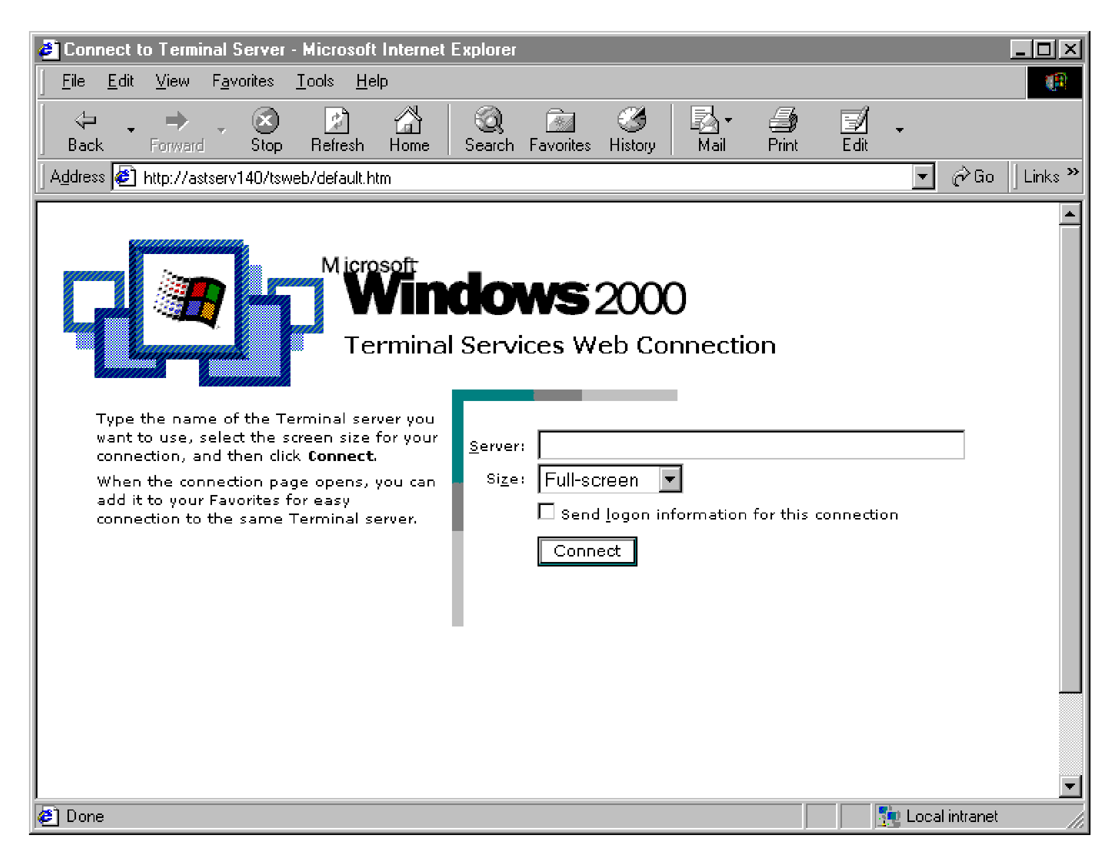
在 Web 服务器上安装 TSAC 之后，用户可以指向缺省登录页，并 （或）传递特定的用户信息来初始化**终端服务器**会话。 TSWeb 目录中安装有三个示例网页。这些网页可以照原样运行，也可以作修改。

备注 如需有关如何配置和使用示例页的详细信息，请参阅《Microsoft® Terminal Services ActiveXÆ 客户端控件部署指南》。

- **Default.htm**。Default.htm 是设计用于从用户端收集**终端服务器**连接信息的示例登录页。您可以通过下面的 URL 来访问该缺省页：

  http://MyWebServer/TSWeb/default.htm

  其中 `MyWebServer` 是 Web 服务器的计算机名或 IP 地址。



- **Connect.asp**。 Connect.asp 是包含实际 ActiveX 客户端控件的示例页，该控件托管终端服务器会话。按照其设计， Connect.asp 不能独立运行，但可以使用下列参数进行调用：

  http://MyWebServer/TSWeb/connect.asp?Server=MyTSServer&Username=MyUser&Domain=MyDomain&rW=80&rH=600

  其中， `MyWebServer` 是 Web 服务器的计算机名或 IP 地址， `MyTSServer` 是**终端服务器**的计算机名， `MyUser` 是一个有效的登录名， `MyDomain` 是定义了该登录名的计算机的名称。

备注 要使用该示例页，必须在 Web 服务器上启用 “活动服务器页”（Active Server Pages，简称 ASP）。

如果通过*防火墙*访问 Internet，确保要打开 TCP 端口 3389。

保护基于 **Web** 的应用程序

Internet 正面临着非法访问日益增长的风险，因此远程控制进程的背后潜伏着许多安全性和可靠性问题。 Internet 是一个公共媒体，任何连接均可能不经意暴露敏感信息，和 （或）令系统遭受恶意行为的蓄意破坏。要充分保护**终端服务器**及它所控制的进程，则应该开发一套充分有效的信息安全 (INFOSEC) 策略。 INFOSEC 策略应包括 “物理安全性”、“网络安全性”、“应用安全性”以及 “安全性跟踪”。

物理安全性

物理安全性涉及服务器及其连接的客户端系统的操作环境。

- 将终端服务器放到一个可免遭物理攻击或远离不利条件的受保护空间内。确保该空间只向经过授权的 （可信任的）人员开放。
- 制订数据备份时间表，并公布有关如何恢复数据的程序。
- 在终端服务器宕机时，立即评估其风险。使用电涌抑制器、不间断电源和备用服务器等硬件保护措施会有助于确保系统持续运行。“网络负载平衡”或具备 “高可用性”的系统可有效减少因部件故障而导致停工的机会。

网络安全性

网络安全性涉及**终端服务器**与客户端计算机之间的数据传输。

- 提供充分的计算机登录安全性。

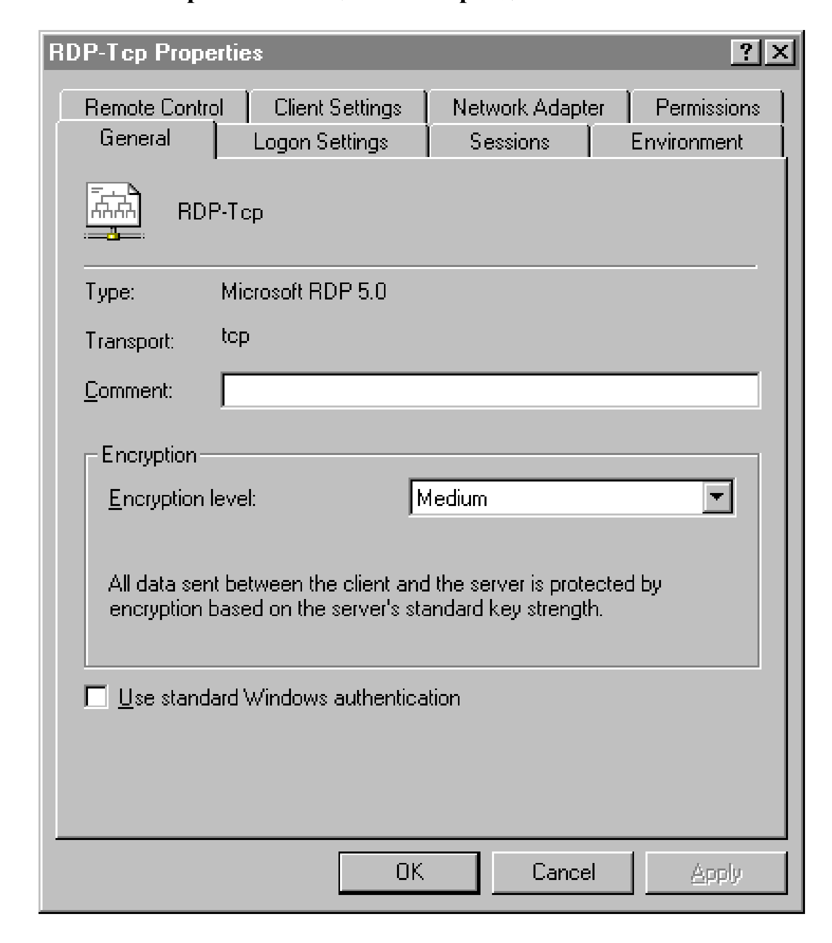
  如需有关详细信息，请参阅 *Terminal Services for In Touch 布署指南*。

- 采用中等 （或更高级的）加密机制。加密可防止欺诈行为，这指的是企图非法截取地址、用户验证、部分或全部传输数据等行为。“终端服务”提供多级加密机制。所有加密级别均采用 RSA RC4 标准加密模式。

级别	描述
低	此级别使用 56 位或 40 位密钥加密从客户端发往服务器的所有数据。在 Windows 2000 客户端连接 Windows 2000 **终端服务器**时，它使用 56- 位密钥，在较早前版本的客户端与它连接时，会使用 40- 位密钥。这种只有输入的加密机制可用于保护敏感数据，如用户口令。
中 （缺省值）	此级别使用 56 位或 40 位密钥对双向传输的数据（从客户端发往服务器，以及从服务器发往客户端）进行加密。Windows 2000 客户端连接 Windows 2000 **终端服务器**时，后者使用 56 位密钥，在更早版本的客户端连接它时，则使用 40 位密钥。使用中等加密机制给在网络上传输以显示于远程客户端的敏感数据进行加密。
高	高级加密方法会影响双向传输的所有数据，并且它使用 128 位密钥进行加密。

要启用加密

1. 在 Windows “任务栏”上单击**开始**，指向**程序**，再指向**管理工具**，然后单击**终端服务配置**。
2. 双击 **RDP-Tcp**。此时会出现 **RDP-Tcp 属性**对话框。



3. 单击**通用**标签以激活**通用**属性页。
4. 选择适当的**加密级别**。

应用程序安全性

“应用程序安全性”是指 InTouch 应用程序、Industrial *SQL Server* 以及其它机密信息系统中内嵌的安全性。

- 使用 $Operator 标记名以提供 InTouch 应用程序内在的安全性。通过给应用程序使用安全设置，您可以将各项功能链接到内部标记名，从而控制允许操作员执行哪些特定的功能。

  如需有关 $Operator 标记名的详细信息，请参阅联机 *InTouch 用户指南*第 2 章的 “使用 InTouch 安全性”部分。

- 使用新的 TseGetClientId() QuickScript 取代 GetNodeName() QuickScript 来识别客户端。在使用 “终端服务”时，GetNodeName() 会返回终端服务器而不是客户端的名称。

- 给 SQL Server 系统管理员 (SA) 帐户添加口令。安装 Industrial SQL Server 时，系统会创建一个全能型 "sa" 登录 ID 及一个空口令。不要使用此帐号来访问数据。相反，请使用缺省的登录 ID （例如 wwUser）。

  如需有关数据库安全性的详细信息，请参阅 *Industrial SQL Server 管理员指南*的 “管理安全性”这章。

安全性跟踪

“安全性跟踪”涉及监视非法入侵企图的能力。如果您怀疑系统遭受任何形式的攻击，则可启用日志功能，以记录一系列可跟踪的事件。通常安全性日志 / 跟踪功能要求使用大量的处理资源，在缺省条件下它是禁用的。因此，我们建议您在开始时只选择监视少量事件。

注意！ “安全性跟踪”要求使用相当多的资源。在评估试验服务器时，切记应启用跟踪功能，否则您可能会低估硬件要求。

要配置跟踪功能，请参阅 “审核策略”，这是 “Windows 2000 本地安全策略”的一部分。

附加信息

如需进一步了解上述内容及相关的安全注意事项，请参阅下列资料：

National Computer Security Center (NCSC) "Rainbow Books" –
http://www.radium.ncsc.mil/tpep/library/rainbow/index.html

Common Criteria (CC) for Information Technology Security Evaluation –
http://www.commoncriteria.org/cc.html

Microsoft Privacy & Security Fundamentals:Security –
http://www.microsoft.com/privacy/safeinternet/security/best_practices/default.htm

Windows 2000 Security Technical Overview –
http://www.microsoft.com/technet/win2000/win2ksrv/sectech.asp

Default Access Control Settings in Windows 2000 –
http://www.microsoft.com/technet/win2000/win2ksrv/technote/secdefs.asp

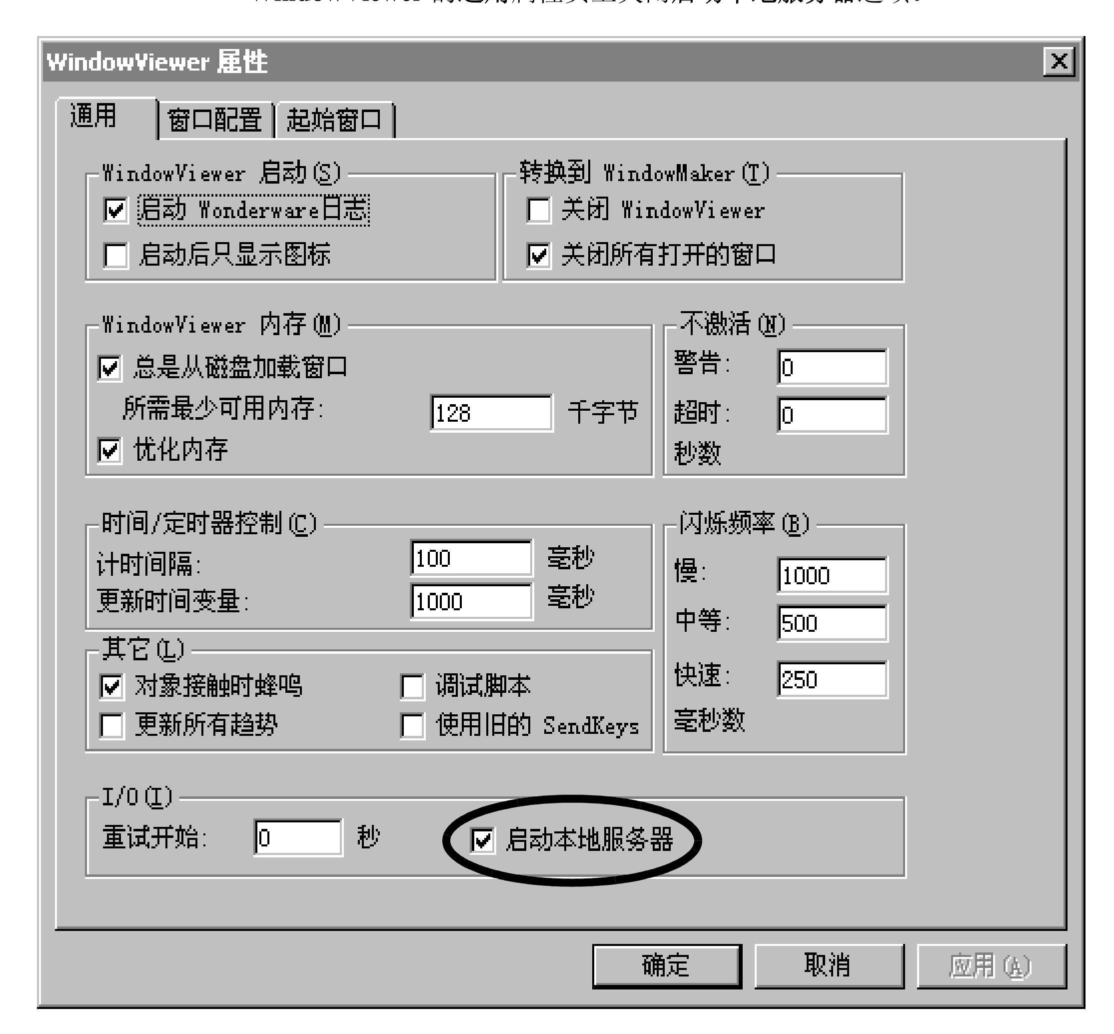
已知问题和限制

下表介绍在**终端服务器**上运行应用程序时的一些限制，以及可能需要实施的建议解决方法。 Wonderware 正积极争取在下一版本中解决其中的大多数问题。

功能	是否支持？	注释
AlarmSuite Logger	否	使用标记名服务器 （仅限于单机）
至 **I/O** 设备或 **MS Office 2000**（例如 **Excel**）的 **DDE**	否	使用标记名服务器 （控制台或单机）。这包括 **DDE QuickScript**：**WWExecute()**、 **WWpoke()** 及 **WWRequest()**
自 **MS Office 2000** 的 **DDE**（例如 **Excel** 中配置的热链接）	是	Excel 和 InTouch 必须在同一会话中运行
历史趋势	是	使用标记名服务器或 NAD 来记录数值。多个会话可能会读取相同的历史文件
InBatch	否	开发中 - 将在未来的版本中支持此项功能
InControl	是	InControl 必须在控制台上运行，并要求使用 v7.1 Patch 03 或更高版本。
InSQL ActiveX 控件 **(OLE DB)**	是	InSQL 应位于单机
InTouch Alarm Logger	否	开发中 - 将在未来的版本中支持此项功能
InTrack OLE 自动化	否	开发中 - 将在未来的版本中支持此项功能
音频 **QuickScript**	是	要求使用 Citrix MetaFrame
打印报警	否	开发中，将在未来的版本中提供支持。您可能需要使用 net use 命令配置并口打印机以便从客户端执行打印任务。如需有关详细信息，请参阅 149 号 “技术说明” (Tech Note)。
保留标记名	是	必须使用 NAD
SPC Pro	否	不支持
SQL Access (ODBC)	是	数据库必须位于单独的计算机上
至 **I/O** 设备或其它 **InTouch** 应用程序的 **SuiteLink**	是	与另一视图会话通讯时，须提供**终端服务器** “节点”名，并将所需会话的 IP 地址追加到 “应用程序”名。例如，view10.103.25.6。

启动本地 I/O

InTouch 不能在**终端服务器**环境下启动 I/O 服务器。为避免在 **WindowViewer** 启动时出现 "正在初始化 I/O" 错误消息，请在 **WindowViewer** 的**通用**属性页上关闭**启动本地服务器**选项。



备注 根据视图会话的启动顺序，您可能需要执行 **IOReinitialize** QuickScript。在启动会从这些服务器读取数值的应用程序之前，切记应确保所有的服务器 (I/O 设备或视图应用程序）均在运行。

执行脚本

终端服务器上运行的所有应用程序均使用相同的定时参考点 (服务器时钟)，所以在发生 CPU 加载异常的情况下，脚本可能不会执行。 CPU 加载异常可能会由视频处理负载过重引起，在多个应用程序定义了相同的脚本触发器 (例如 End-of-Shift 事件）时，也可能会发生此种情形。因此，如果服务器正忙于处理许多客户端的脚本，则在定时器通常可以启动脚本的时间间隔之内，它可能会无法启动其它客户端上的脚本。这可能会导致客户机上的脚本不执行。

为确保正确执行脚本，请将使用公用触发器的脚本合并到一起，并将它们移植到一个应用程序中，例如标记名服务器。这是执行试验布署的主要原因之一。通过试验布署，您可以有机会执行各种 “因果型” 试验方案，并确定所选硬件配置是否够用。

服务器硬件要求

Terminal Services for InTouch 必须安装在新的 **Windows 2000 Server** 或 **Advanced Server** 上。它不能从 Windows NT 系统升级。下表提供了建议采用的硬件平台。对于典型的 InTouch 应用程序，它们可以提供良好的性能。

建议配备的硬件

CPU [1]	物理内存 [2、3]	虚拟内存 [4]	客户端数
奔腾 III 450 MHz	384 MB	960 MB	5
奔腾 III 500 MHz	1024 MB	2560 MB	15
奔腾 III 700 MHz	2048 MB	5120 MB	25

1. 采用多个处理器可以改善性能。
2. 对于 Windows 2000 Advanced Server，须增加 128 MB 的 RAM。
3. 内存要求取决于应用程序负载与连接的用户数。 RDP 将要求运行 InTouch 的每位用户使用 40-60 MB，而 ICA 的要求则稍低一些。
4. 虚拟内存 （页面文件大小）应该是物理内存的 250%。

备注 要估计一台服务器可支持多少个用户，一个较好的方法是让两至五个用户在系统上试运行，以测试系统性能，然后比较各个结果。

如需有关分析系统性能的详细信息，请参阅 *Terminal Services for In Touch 部署指南*。

硬盘空间

一个或多个硬盘，其中包含系统文件的分区至少应配备 2 GB 空间。

通过使用 “冗余廉价磁盘阵列”（Redundant Array of Inexpensive Disk，简称 RAID），有助于防止丢失数据并降低服务器停工时间。

联网

10/100Mbps 网卡。使用 TCP/IP 协议的网络。

其它驱动器

CD-ROM 驱动器

使用高密 3.5 吋磁盘驱动器作为 A 驱，除非：

- 计算机支持从光盘启动安装程序。

- 您通过网络安装 Windows 2000。

附件

键盘、鼠标 （或其它定位设备）及显示器 （VGA 或更高性能）。

不间断电源 （Uninterrupted Power Supply，简称 UPS）

备注 在安装 Windows 2000 之前，请检查在 Windows 2000 “兼容硬件表”（Hardware Compatibility List，简称 HCL）中是否列有您的硬件。Microsoft 只对 Windows 2000 HCL 中列出的设备提供经测试的驱动程序，因此使用未在 HCL 中列出的硬件可能会在安装期间或之后导致问题。您可以在 Internet 上找到最新版本的 HCL，它位于 http://www.microsoft.com/hwtest/hcl。

检查影响性能的外设

外设也可能会影响运行 “终端服务”的服务器的性能：

- 硬盘磁盘速度对终端服务器的性能而言非常重要。小型计算机系统接口 (SCSI) 磁盘驱动器 （特别是与 Fast SCSI 和 SCSI-2 兼容的设备）的输入输出性能明显优于其它类型的驱动器。
- 网络适配器。建议使用高性能的网络适配器，特别是用户需要访问网络服务器上存储的数据，或访问客户端 / 服务器应用程序 （如 Wonderware InTouch）时。使用多个网络适配器可显著提高网络输入输出性能。

RDP 客户端硬件要求

运行“终端服务”的客户端不需要拥有很强的处理能力。例如，配备 32 MB 的 RAM 和 VGA 显卡的奔腾机就足够了。因此可以很容易地将 “终端服务”集成到使用较旧计算机和设备的网络。

备注 使用标准 VGA 显卡可能会限制显示大小和颜色深度。

根据所使用的客户端的型号，您需要注意某些性能方面的问题。例如，使用 Microsoft Windows CE 作为操作系统的设备的速度不比使用 Linux 的相同设备快。

规划终端服务安装

“终端服务”安装的成功关键是正确规划。在安装 “终端服务”之前，请执行下列任务：

- 确定您需要安装在服务器上的客户应用程序 （例如 Wonderware InTouch）。

- 确定客户端的硬件要求。
- 确定支持客户端所需的服务器配置。
- 确定 “终端服务”以及要运行的所有其它应用程序所需的许可证。

在安装 “终端服务”之前，确定要在客户端桌面上部署的应用程序。大多数可在 Windows 2000 上正确运行的应用程序都可以在 “终端服务器”上运行。不过，有些应用程序可能需要在局部修改之后，才能在 “终端服务”环境下成功运行。在运行环境下部署这些应用程序之前，请先在测试服务器上安装这些应用程序，以确保它们与现有的应用程序兼容。

如需有关详细信息，请参阅 *Terminal Services for InTouch 部署指南*。

安装终端服务

在安装 Windows 2000 Server 期间，您可以在服务器上安装 “终端服务”，也可以在 “安装”之后，通过 “控制面板”中的**添加 / 删除程序**来安装 “终端服务”。

重要！ 终端服务必须在安装任何软件产品之前安装。软件产品必须在 “安装模式”下添加到终端 “服务器”。使用 Windows “控制面板”中的**添加 / 删除程序**可以自动激活 “安装模式”。这样您便可以为机器上的所有用户配置软件，而不只限于原来安装程序的用户。

终端服务组件

“终端服务”的安装包括下列组件和选项：

- **终端服务**。提供多会话环境，供客户端访问服务器上基于 Windows 的应用程序。“终端服务”包括两个子组件：
- **客户端生成器文件**。它包含一个为 “终端服务”客户端创建安装磁盘的向导。
- **启用终端服务**。在服务器上启用 “终端服务”软件。您可以使用此选项来启动和停止 “终端服务”。
- **终端服务授权**。将计算机配置为给客户端提供许可证的 “终端服务”授权服务器。

要安装终端服务

1. 在 Windows “任务栏”上，单击**开始**，指向**设置**，然后单击**控制面板**。
2. 双击**添加 / 删除程序**图标。此时会出现**添加 / 删除程序**对话框。

3. 单击**添加 / 删除 Windows 组件**。此时会出现 **Windows 组件向导**。
4. 选择**终端服务**和**终端服务授权**选项，然后单击**下一步**。（“终端服务”授权可能会安装到单独的一台机器上）。
5. 选择**应用程序服务器模式**选项，然后单击**下一步**。
6. 选择**跟 Windows 2000 用户兼容的权限**选项，然后单击**下一步**。
7. 选择**您的域或工作组**选项，并指定授权服务器数据库的目录位置

备注 仅当选定**终端服务授权**选项时，此选项才会出现。

8. 单击**下一步**开始安装。

安装完成之后，**管理工具**菜单中会添加多个项目。下表介绍这些新增的项目：

项目	描述
终端服务客户端生成器	为安装 “终端服务”客户端软件创建软盘。
终端服务配置	管理 “终端服务”协议配置和服务器设置。每个网络适配器只能配置一个 RDP 连接。
终端服务授权	管理 “终端服务”的客户端访问许可证。
终端服务管理器	管理和监视运行 “终端服务”的服务器上的会话和进程。

安装 Terminal Services for InTouch

要安装 Terminal Services for InTouch，您必须使用内置的 Administrator（管理员）帐户登录到运行 “终端服务”的服务器。

备注 如果已经安装了 “非终端服务版”的 InTouch，则无法升级到 “终端服务”版的 InTouch。不过，您可以从先前安装的 InTouch “终端服务版”升级到最新的版本。

要安装 Wonderware Terminal Services for InTouch

1. 将 Wonderware Terminal Services for InTouch 光盘插入 CD-ROM 驱动器，等待自动运行程序自动激活 InTouch **SETUP.EXE** 程序。
2. 此时会出现**欢迎**对话框。单击**下一步**。

3. 依照屏幕上的说明进行操作以便安装 InTouch。

4. 在安装 “系统文件” 和 “公用” 组件之后，安装程序会提示您重新启动系统。单击**确定**以重新启动计算机。在系统重新启动之后，InTouch 安装程序将自动继续执行。

备注 如果此系统上已经安装有 Terminal Services for InTouch，则不会如上面的步骤 4 所提到的那样提示您重新启动系统。

---

5. 在成功安装 InTouch 之后，会出现一个对话框，告诉您已经完成安装。

6. 此外，在成功安装 InTouch 后，还可能会出现下面的**问题**对话框：



单击**是**以继续安装 MSDE 2000，或单击**否**以退出 InTouch 安装程序。

备注 Microsoft 建议在 “终端服务器” 节点之外的其它节点上安装 SQL Server/MSDE。如需有关如何设置 SQL Server/MSDE 节点的详细信息，请参阅 《InTouch 终端服务布署指南》或 《InTouch 报警布署指南》。

安装 InTouch 时，您可以使用三个选项：**整个开发系统**、**只有运行时**或 **Factory Focus**。此选项确定了**终端服务器**及其相应的客户端运行 InTouch 的方式。您不能混合使用这两种 InTouch 配置，也不能让它们完全相同。例如，您可能希望让一些客户端运行 Runtime，让其它客户端运行 Factory Focus，此时您便需要两套**终端服务器**。

---

7. 安装适当的 Wonderware 许可证。

   现在您就可以开始使用 Terminal Services for InTouch 了。

在终端环境下测试应用程序

为确保应用程序能够在 “终端服务” 环境下正确安装和运行，在该种环境下进行测试是关键。因而有必要构建一个典型的使用方案，确保使用适当数目的会话运行应用程序一段时间，以模拟实际应用。

Terminal Services for InTouch QuickScript 函数

本节向您介绍可供 “终端服务”使用的三个 InTouch QuickScript 函数。

TseGetClientId()

如果 View 应用程序在 “终端服务器”客户端上运行，则返回字符串形式的客户端 ID （客户端的 TCP/IP 地址）。否则它会返回一个空字符串。

语法 *MessageResult*=TseGetClientId();

参数	**描述**
MessageResult	显示客户端 ID 的 “消息”型标记名。

附注 此 ID 供内部使用，以生成 SuiteLink 服务器名及 logger 文件名。

示例 MsgTag=TseGetClientID();
客户端 IP 地址，例如， 10.103.202.1 会被返回给 MsgTag。

TseQueryRunningOnConsole()

如果 View 应用程序在 “终端服务器”控制台上运行，则返回一个非零的整数值。否则它会返回零 (0)。

语法 *Result*=TseQueryRunningOnConsole();

参数	**描述**
Result	如果 View 未在 TS 控制台上运行，则返回 0。

附注 无

示例 IntTag=TseQueryRunningOnConsole();
如果 WindowViewer 在 “终端服务器”控制台上运行，则返回 IntTag = 1。

TseQueryRunningOnClient()

如果 View 应用程序正在 “终端服务器”客户端上运行，则返回一个非零的整数值。否则它会返回零 (0)。

语法 *Result*=TseQueryRunningOnClient();

参数	**描述**
Result	如果 View 未在 TS 客户端上运行，则返回 0。

附注 无

示例 IntTag=TseQueryRunningOnClient();
如果 WindowViewer 在 “终端服务器”客户端上运行，则返回 IntTag = 1。

建立终端会话

在完成服务器和客户端配置之后，您就可以建立“终端”会话。您可以从客户端访问网络和本地资源，包括硬盘和打印机。在结束会话时，您可以断开连接并稍后重新建立会话，也可以注销以完全关闭会话。

连接终端服务器

连接“终端服务器”时，您可以选择不同的选项，以适应低速网络并提高会话性能。

要连接终端服务器

1. 启动“终端服务客户端”。此时会出现**终端服务客户端**对话框，它包含下列选项：

选项	操作
服务器	输入“终端服务器”的名称或 TCP/IP 地址
屏幕区域	选择屏幕分辨率。此设置与服务器的屏幕分辨率无关。
可用服务器	从可用服务器列表选择服务器。
低速连接	如果您使用调制解调器或低速网络，请单击“低速连接”。
缓存位图到磁盘	单击“缓存位图到磁盘”，以便将桌面显示元素保存到本地缓存中。此选项可以从本地缓存刷新屏幕，从而提高性能。

2. 单击**连接**。
3. 登录到“终端服务器”。

结束终端会话

“终端服务”提供了两个选项供您去结束“终端”会话：

- 断开会话连接。断开连接，并让会话继续在服务器上运行。您可以稍后重新连接到服务器并继续该次会话。例如，如果您在服务器上执行比较耗时的任务（如在数据库上执行查询），则可以在启动任务后断开会话连接。稍后，您可以再次登录到服务器，继续该次会话并继续执行任务或检查结果。
- 注销会话。注销会话可以结束当前在服务器上运行的会话。会话内运行的任何应用程序都会关闭，所有未保存的数据均将丢失。切记要注销会话，以便将服务器资源让给新会话，这点非常重要。

其它应用程序问题

有些应用程序所具有的功能可能无法支持 “终端服务”或导致它们性能低下。例如，只有在下列情况下，才可以将要求使用特殊硬件 （如条形码扫描仪或智能卡阅读器）的应用程序用在 “终端服务”客户端上：

- 这些设备已连接到计算机或终端，并且外设被识别为键盘类设备。

连接软件和硬件支持与客户端的连接。

配置客户端设置

为确保系统资源可供活动的 “终端会话”使用，您可以为断开连接和闲置的会话设置时间限制。

在服务器 “终端服务配置” **RDP-Tcp 属性**对话框的**会话**属性页上，您可以给会话指定时间限制。下表介绍用于限制会话长度的一些设置。

设置	描述
结束断开连接的会话	指定断开连接的会话可保留的最长持续时间。会话将会被重置，并且在时间限制过后便不能再恢复。
活动会话限制	指定连接的最长持续时间。达到时间限制时，会话会被断开，让会话继续在服务器上运行或重置会话。
闲置会话限制	指定在会话断开连接或重置前允许闲置的最长时间 （无连接活动的时间）。

第 15 章

InTouch Application Publisher

本章综述 InTouch Application Publisher，您可以用它来创建自解压文件，包含所需的全部相关文件及安装程序，以最大程度减少通过 Internet 进行下载所需要的时间。

目录

发布 InTouch 应用程序

Application Publisher 创建一个自解压文件，包含在另一个 InTouch 节点上安装应用程序所需的全部相关文件及安装程序。 Application Publisher 会压缩 “应用程序”来最大程度减少通过 Internet 进行下载所需要的时间。

备注 您可以将这些压缩过的应用程序复制到软盘、 zip 盘等，以便在其它节点上安装。

Application Publisher 的压缩率约为 19:1。此种压缩率可以将 20 MB 的完整开发程序压缩到一张标准软盘上。如果该文件只是运行时文件，则可以将 30 到 40 MB 的应用程序压缩到一张标准软盘。

在发布应用程序之后，您可以在一个或多个其它 InTouch 7.1 （或更高版本）节点上重建 InTouch 应用程序。

您可以使用多种方法将已发布的应用程序文件加载到另一个 InTouch 节点上：

- 复制到软盘
- 通过公司内部网 (Intranet) 复制文件
- 通过 Internet 复制文件
- 通过在目标节点上将自解压文件当作程序来运行，就可以使用重建后的 InTouch 应用程序。

要发布 **InTouch** 应用程序

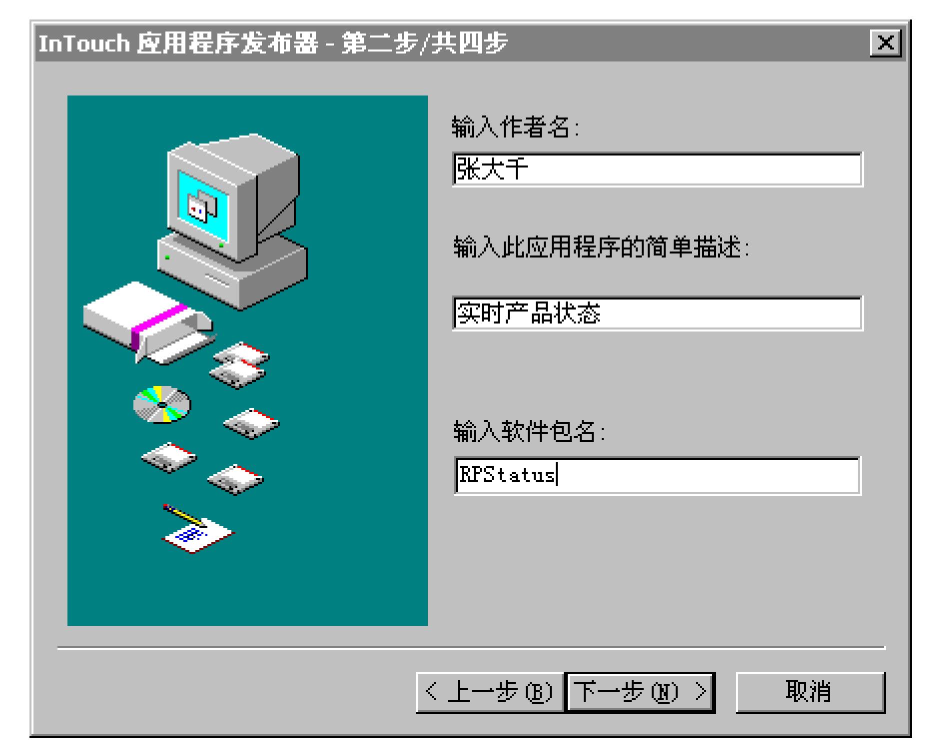
1. 在 WindowMaker 中，从 “应用程序浏览器” 下打开应用程序。



2. 选择 **Application Publisher**。此时会出现 **InTouch Application Publisher** - 第一步 / 共四步对话框。



3. 单击下一步。此时会出现 **InTouch Application Publisher - 第二步 / 共四步**对话框。



4. 在**作者名**框中，输入有关应用程序联系人的姓名 （最多 256 个字符）。（这是一个可选输入项）。例如：

   Ernie Tankhammer

   Ernie Tankhammer 高级工程师电话：1-800-555-1234

5. 在**描述**框中，输入应用程序的描述 （最多 256 个字符）。（这是一个可选输入项）。例如：

   实时生产状态

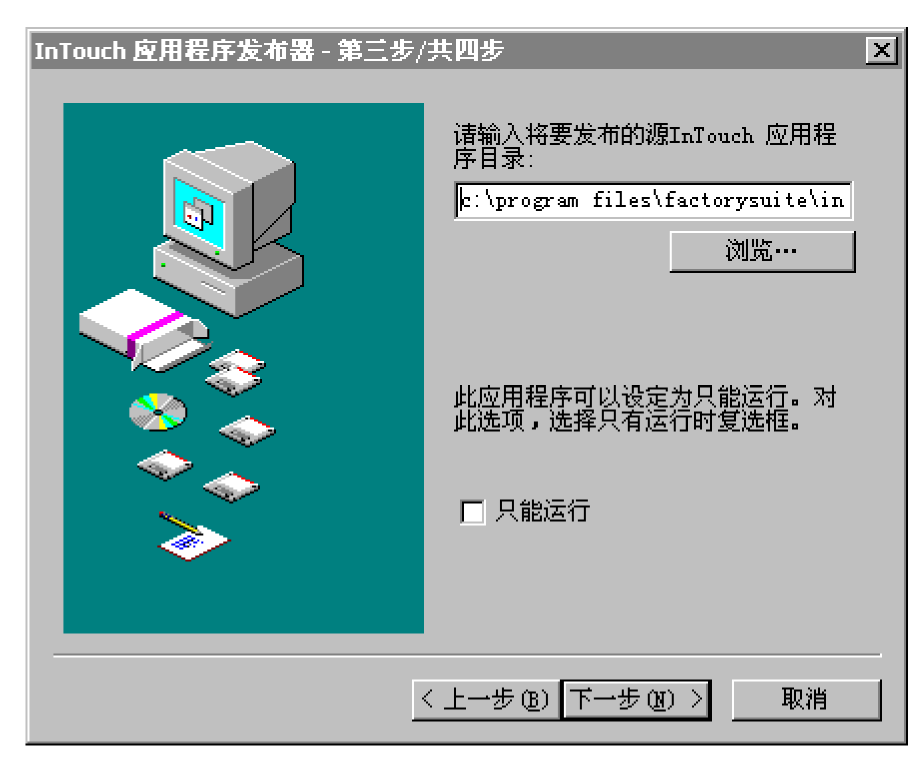
   5 号工厂的 3 号工作区，要求访问 3 号区 FactorySuite 文件服务器以便访问数据点。

6. 在**软件包名**框中，输入软件包的名称 （最多 32 个标准 Windows 文件命名字符）。您**必须**输入**软件包名**。

备注 **软件包名**是用于识别已发布的应用程序的唯一名称。它用于维护相同的已发布应用程序的可同时下载的版本。

---

7. 单击下一步。此时会出现 **InTouch Application Publisher - 第三步 / 共四步**对话框。



8. 在输入框中，输入 InTouch 应用程序源的路径。缺省值是 WindowMaker 应用程序目录。

9. 选择**只能运行**选项，以防在已发布的文件中包括 WindowMaker 的开发文件。

10. 单击**下一步**。此时会出现 **InTouch Application Publisher - 第四步 / 共四步**对话框。



11. 确认第一个框中的可执行文件名是否正确。（缺省条件下，这与您指定的**软件包名**相同）。
12. 在第二个框中，输入用于保存已发布文件的目录路径。（缺省条件下，这会保存到当前的 Temp 目录）。或者，单击**浏览**以找到不同的目录。

> **备注** 在上例中，文件会发布到公司内部网上的 FactorySuite Web 服务器。

13. 单击**完成**。

用多种分辨率发布应用程序

您可以用多种分辨率发布应用程序。这会消除客户端无法以正确的分辨率查看应用程序的问题。

要使用不同的分辨率发布应用程序

1. 在运行 WindowMaker 时，改变个人计算机的屏幕分辨率。
2. WindowMaker 会通知您分辨率已改变，并提示您接受改变，单击**确定**。

3. 从“应用程序浏览器”中，打开**应用程序**，然后运行 Application Publisher。

4. 更改应用程序的**软件包名**。例如，将 **Dairy** 更改为 **Dairy_2**。

备注 如果要保持**软件包名**不变，而只改变文件名（缺省条件下，它们是相同的），请在 **InTouch Application Publisher - 第四步 / 共四步**对话框中更改文件名。

要以多种分辨率发布相同的应用程序，则**必须**改变文件名。如果文件名相同，则它只会覆盖原来的应用程序。

发布到 Web 服务器时，它会显示成：

描述	Dairy 处理应用程序
发布者	Navin Johnson
文件名	Dairy.exe / 视频分辨率 ...(1024x768)

描述	Dairy 处理应用程序
发布者	Navin Johnson
文件名	Dairy_2.exe / 视频分辨率 ...(800x600)

附 录 A

InTouch Windows NT 服务综述

服务指的是 Windows NT 中的一个进程，它无需用户干预就可执行特定的后台系统功能。您可以将服务配置为在每次应用程序所处的计算机启动时自动启动。此外，服务启动时也不会影响 Windows NT 的安全系统。本附录将介绍 InTouch （7.1 或更高版本）的 Windows NT 服务。

InTouch 提供四种 Windows NT 服务：

InTouch 服务	Windows NT 服务缺省路径
Wonderware Logger	C:\Program files\Factorysuite\Common\Wwlogsvc.exe
Wonderware SuiteLink	C:\Program files\Factorysuite\Common\Slssvc.exe
Wonderware NetDDE Helper	C:\Program files\Factorysuite\Common\Wwnetdde.exe
Wonderware WindowViewer	C:\Program files\Factorysuite\Intouch\View.exe

备注 如需有关其它 FactorySuite 组件的 Windows NT 服务的详细信息，请参阅 *Wonderware FactorySuite 管理员指南*。

为何使用 Windows NT 服务？

InTouch Windows NT 服务可提供下列功能：

操作员登录和注销。

在正常的工厂运行期间，操作员通常要在换班时登录和注销 Windows NT 操作系统。在 WindowViewer 作为 Windows NT 服务运行时，即使不同的操作员进行登录和注销，它仍可以继续记录历史数据、收集报警信息、处理脚本、充当 I/O 服务器，以及作为 I/O 客户端来写入数值。

断电损失。

如果发生供电中断，在恢复正常的工厂运行之前，操作员通常要等待一段时间。大多数灾难恢复计划均要求在恢复供电之后，立即启动关键的计算机系统。 Microsoft Windows NT WorkStation 和 Windows NT Server 在恢复供电之后均可自动重新启动。由于 InTouch 组件软件是作为一套 Windows NT 服务运行的，因此关键的系统可以在不影响 Windows NT 安全性的前提之下立即开始运行。

持续运行。

现代工厂将资源效率视为主要的考虑因素。为获得最高水平的生产能力，许多工厂都使用持续运行环境。在 InTouch 组件作为 Windows NT 服务运行时，工厂可以保证一周七天、一天 24 小时连续作业。

在桌面和系统环境下运行 Windows NT 服务

启动 Windows NT 操作系统时，配置成自动启动的服务会在 “后台”启动，而不会在桌面上显示图形用户界面 （Graphical User Interface，简称 GUI）。在这种情况下，服务是在*系统环境*下运行的。操作员或其他用户登录到系统之后，任何在系统环境下运行的具有关联 GUI 的服务会在用户登录时自动显示在桌面上。在此种情况下，服务是在*桌面环境*下运行的。

因此，如果您将 WindowViewer 服务配置为自动启动，则在 Windows NT 系统启动后，该项服务会立即在系统环境下运行。随后当用户登录到系统时， WindowViewer 服务会继续运行，但此时是在桌面环境下运行的（WindowViewer 的 GUI 会自动出现）。

备注 如果您在定义 InTouch “访问名”时启用了**只提示激活项**选项，并且定义了只在特定的 InTouch 应用程序窗口中激活的 I/O 标记名 （并非用在应用程序所有地方的标记名），则可以取消激活这些标记名。在注销系统之后， WindowViewer 尽管仍在运行，但不会显示 GUI，因此您无法使用键盘、鼠标或其它定位设备来浏览应用程序。不过，如果使用 QuickScript 关闭应用程序窗口，则该窗口会自动从内存卸载，从而终止与这些标记名的链接。

创建主用户帐户

当启动 Windows NT 操作系统服务时，它将在一个称为*本地系统帐户*的用户环境下运行。配置为自动启动的每个 Windows NT 服务均在此本地系统帐户下运行。当您使用自己的帐户登录到 Windows NT 操作系统并启动应用程序时，则系统将在您的用户帐户下运行。

本地系统帐户在 Windows NT 安全系统中具有很少的权限，因而不能保证登录到系统的用户有权运行 InTouch 服务。这包括访问文件、与其它程序和服务交换数据以及修改注册表等操作。 InTouch 要求用户帐户在 Windows NT 安全系统中具有比本地系统帐户更高的访问级别，从而使服务可以执行这些类型的操作。

要实现这一点，您必须创建一个 InTouch **"主"用户帐户**。（建议用户帐户名为 "wwservices"）。此主帐户应具有本地计算机的 "管理"权限，以便有足够的权限来启动 InTouch 服务。随后在 InTouch 安装期间，当出现 FactorySuite **公用组件**对话框时，您可以输入本地计算机的节点名（不是域名），以及主帐户的用户名和口令。

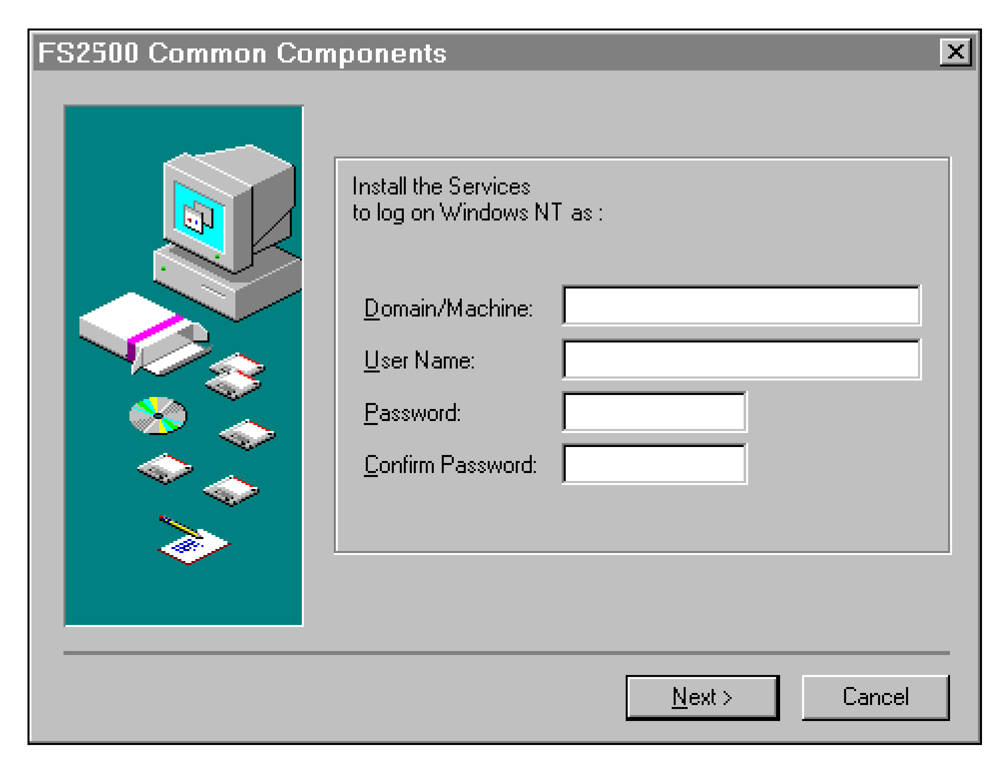
备注 要对主帐户进行任何更改，必须在安装 InTouch 期间进行或通过运行 "Wonderware 服务用户"程序 WWUSER.EXE 来完成。

在设置主帐户、登录到计算机并启动 InTouch 组件（例如 WindowViewer）之后，该组件的服务即在主帐户而不是用户帐户下运行。这就确保了您具有必要的权限来启动 InTouch 服务。

备注 在讨论 Windows NT 服务时，可能使用*模拟帐户*一词而不是主帐户。模拟帐户是提供对您的站点或服务器的限制资源 "区域"访问的 "用户 / 用户组"帐户。在 Windows NT 中，它称为 "NT 用户 / 用户组帐户"，必须由 "系统管理员"设置。

配置系统权限

在 InTouch 安装期间，系统会提示您为管理帐户输入用户名和口令。此信息通常用于设置您的 NT 用户模拟帐户。 Wonderware 服务，如 Wonderware NetDDE Helper 和 Wonderware WindowViewer，使用此信息来自动登录，并且在系统自动启动期间自动启动相应的服务。



要配置系统权限

1. 在**域 / 机器**框中，请输入系统域名或节点名。
2. 在**用户名**框中，输入您的用户标识。
3. 在**口令**框中，输入您的系统口令。

备注 您提供的**用户名**和**口令**必须是通过 “NT 用户管理器”配置的有效 “管理员”级登录。

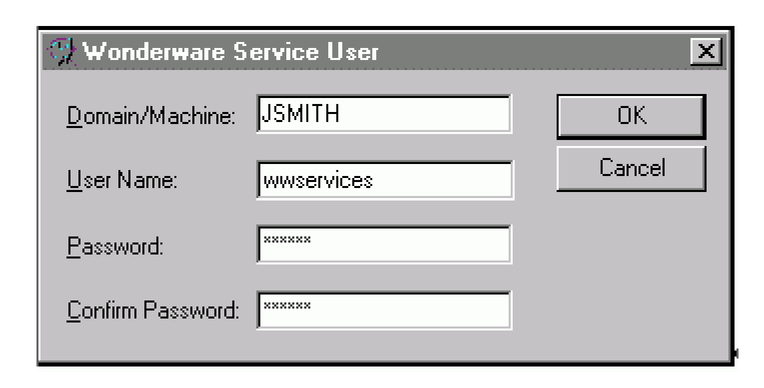
4. 在**确认口令**框中，请重新输入系统口令以进行确认。

提示 安装之后，如果您需要修改此信息，请运行安装目录下的 “Wonderware 服务用户”程序 (wwuser.exe)。例如 \Program Files\FactorySuite\Common。

如果您已安装 InTouch 但需要修改安装程序，以便使用新的 “主”用户帐户，请参阅下面有关如何设置新 “主帐户”的步骤。

要设置 FactorySuite 服务以使用新的主帐户：

1. 运行 “Wonderware 服务用户”程序。（缺省条件下， wwuser.exe 位于目录 C:\Program Files\Factorysuite\Common 中）。此时会出现 **Wonderware** 服务用户对话框。



备注 在此处输入的信息必须与当前主帐户相符。

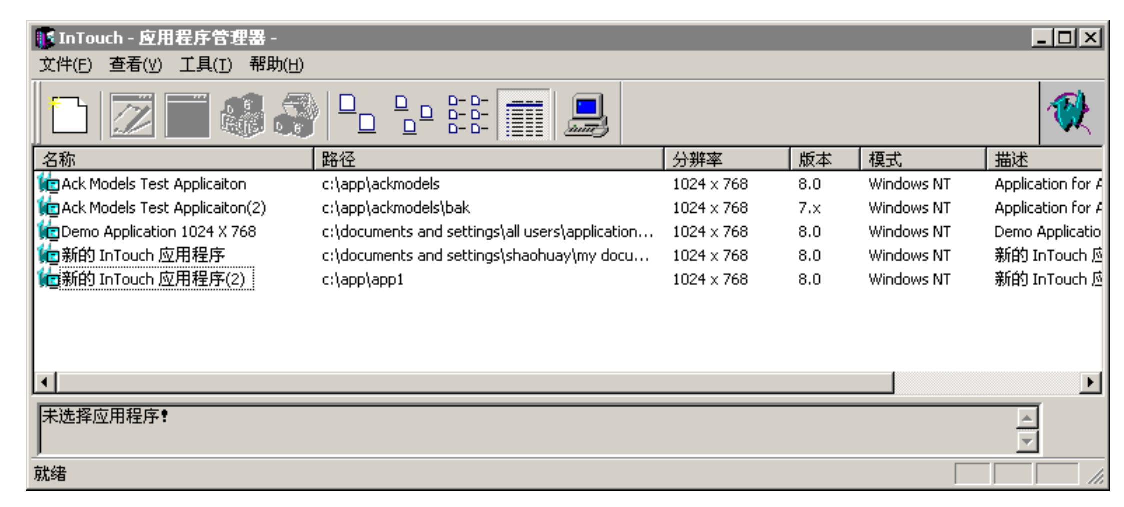
2. 在**域 / 机器**框中，输入机器名。
3. 在**用户名**框中，输入新 “主帐户”的名称。
4. 在**口令**框中，输入口令。
5. 单击**确定**。

配置 InTouch 服务

InTouch 服务 **Wonderware Logger**、 **Wonderware SuiteLink** 以及 **Wonderware NetDDE Helper** 等，均在 FactorySuite 安装期间进行安装并配置为自动启动。不过，您必须将 **WindowViewer** 配置为可以自动启动的 Windows NT 服务。

要将 WindowViewer 配置成作为 NT 服务启动

1. 启动 InTouch 程序 (intouch.exe)。此时会出现 **InTouch** 应用程序管理器对话框。



2. 单击**节点属性**工具，或者在**工具**菜单上，单击**节点属性**。此时会出现**节点属性**对话框，并显示**应用程序开发**属性页。

> **提示** 要快速访问**节点属性**对话框，请右击显示窗口的空白区，然后单击**节点属性**。



3. 选择**启动 WindowViewer 作为 NT 服务**选项。

备注 作为 Windows NT 服务启用 WindowViewer 时，“网络应用程序开发”会被禁用。

---

4. 单击**确定**。

备注 如果将 WindowViewer 配置为 NT 服务，并随后直接启动它（使用它的图标或 Windows 开始菜单等），则在大约 15 秒钟的延迟之后，WindowViewer 会显示一个窗口。此项延迟是 WindowViewer 尝试连接“NT 服务控制管理器”造成的。如果未能连接到“服务控制管理器”，则 WindowViewer 会显示下面的消息框：



如果单击**是**， WindowViewer 将作为应用程序，而不是 NT 服务来启动。如果单击**否**，用于启动 WindowViewer 的命令将被取消。

提示 您还可以使用其它一些方法让 WindowViewer 作为 NT 服务启动：

1. 在 WindowMaker 中，单击菜单栏右上角的 Runtime!FastSwitch。

2. 从“InTouch 应用程序管理器”启动 WindowViewer。

3. 要在命令提示符下启动 WindowViewer 服务，请使用下面的命令：

   Net Start View

4. 要在命令提示符下停止 WindowViewer 服务，请使用下面的命令：

   Net Stop View

注意！ 如果 WindowViewer 配置成以 NT 服务方式自动启动，则系统重新启动时，上次由“InTouch 应用程序管理器”使用的 InTouch 应用程序会自动启动。不过，如果正在编辑多个 InTouch 应用程序，则掉电或重新启动系统后，可能会启动错误的应用程序。

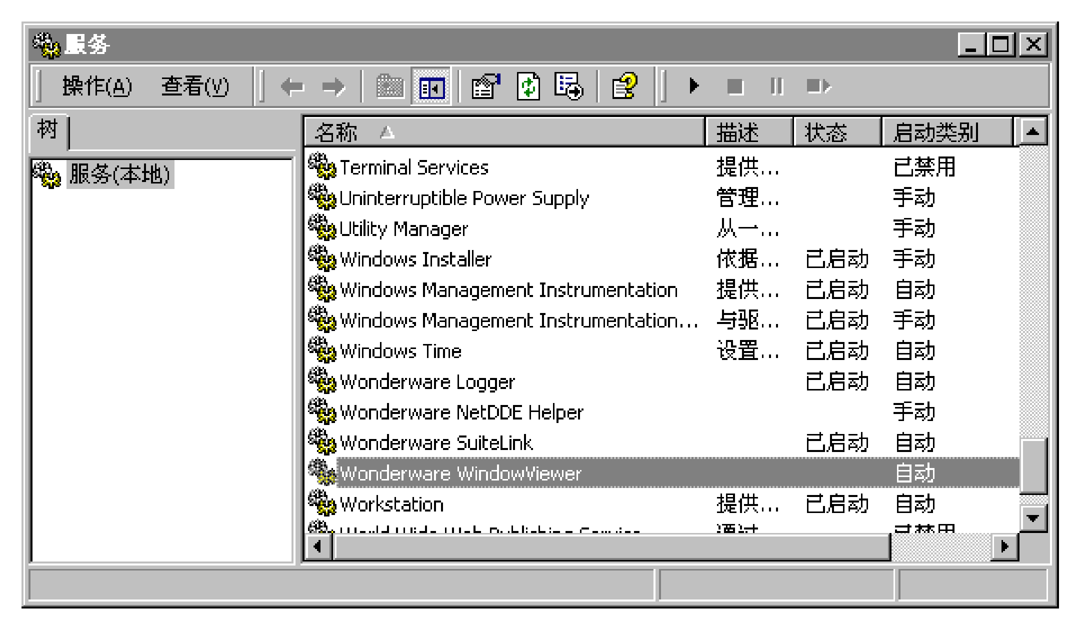
要停止运行 WindowViewer 服务

通过在**节点属性**对话框中关闭**启用 WindowViewer 作为 NT 服务**选项，可以停止 WindowViewer 作为 NT 服务运行。

通过关闭计算机，您可以停止运行 WindowViewer 或任何其它服务。您还可以通过 Windows “控制面板”来停止 （或启动）任何服务。

要通过控制面板停止 / 启动 NT 服务

1. 打开 “控制面板”，双击**服务**图标。
2. 此时会出现**服务**对话框。



3. 在 “服务”列表上辉亮显示要停止 （或启动）的服务，然后单击**停止**或**启动**，再单击**关闭**。

手动安装 / 删除 NT 服务

通常， InTouch 的 Windows NT 服务在 FactorySuite 安装过程中自动安装，通过双击 “控制面板”中的**添加 / 删除程序**图标可以将它们删除。不过，如果需要手动安装或删除 InTouch Windows NT 服务 （例如 Wonderware WindowViewer），请执行以下步骤。

要安装 Windows NT 服务

1. 从 “开始”菜单中选择**运行**，或打开一个 “MS-DOS 命令提示符”窗口。
2. 输入此命令：

```
"C:\Program Files\FactorySuite\InTouch\view.exe" -
   install
```

要删除 Windows NT 服务

1. 从“开始”菜单中选择**运行**，或打开一个“MS-DOS 命令提示符”窗口。
2. 输入此命令：

```
"C:\Program Files\FactorySuite\InTouch\view.exe" -
   remove
```

备注 这两个命令是安装和删除 Wonderware WindowViewer 服务的示例。对于其它的 InTouch 服务，路径名将包括目录 **\Common** 而不是 \InTouch。

如果在**运行**对话框中输入命令，请确保用英文双引号 (" ") 括起该命令。例如 "Program Files"。如果删除某项服务，则 Windows NT 会首先停止该项服务，然后再删除它。删除服务时，不会删除该应用程序的 .exe 文件。

配置 NT 服务启动选项

要配置 InTouch 服务的启动选项

1. 打开 Windows “控制面板”。
2. 双击**服务**图标。此时会出现**服务**对话框。
3. 从“服务”列表中选择所需的 InTouch 服务，然后单击**启动**。
4. 此时会再次出现**服务**对话框。



5. 在**启动类型**组中，选择要使用的选项，然后单击**确定**。

选项	描述
自动	Windows NT 重新启动时，服务可以自动启动而无需用户干预。
手动	用户或应用程序进程必须明确启动服务。
禁用	禁止服务启动。（通常保留作疑难排解之用）。

备注 Wonderware 建议给所有的 InTouch 服务使用缺省启动选项。

各项 InTouch 服务之间的相关性

安装服务时，安装程序会给 Windows NT 操作系统提供一份相关性列表。如果某项服务要在其它服务启动之后才能启动，则 Windows NT 在启动该项服务*之前*，会先执行检查以确定其它服务正在运行。根据运行 WindowViewer 的要求，您应该了解下列相关性。

如果您打算使用 “分布式报警” 或 “分布式历史”，或需要访问 Network DDE 数据，则必须运行 Wonderware NetDDE Helper 服务。Wonderware NetDDE Helper 服务还依赖于要安装的 Network DDE 和 Network DDE DSDM 服务是配置为 “手动” 还是 “自动” 启动。在安装过程中， Wonderware NetDDE Helper 服务会被配置为 “手动” 启动，这表示 WindowViewer 会在启动时自动启动此项服务。

如果需要将 WindowViewer 用作 SuiteLink 服务器或客户端，则必须运行 Wonderware SuiteLink 服务。 Wonderware SuiteLink 服务还要求已安装 Microsoft TCP/IP。

如果要在运行 WindowViewer 时存储任何消息或错误，则必须确保已安装 Wonderware Logger 服务。作为 FactorySuite 公用组件安装的一部分，必须安装 Wonderware SuiteLink 和 Wonderware Logger 服务并将它们配置为在启动期间**自动**启动。

如需有关详细信息，请参阅 “配置 NT 服务启动选项”。

注销 Windows NT

注销 Windows NT 操作系统时，它会发送控制注销消息 **wm_close** 和 **wm_terminate**，以关闭当前在桌面环境下运行的所有应用程序。缺省条件下，需要安装 Microsoft Network DDE 并将它配置为 “监听” 这些注销消息。如果收到任何消息， Microsoft Network DDE 会自动终止打开的网络连接，并且在注销系统时， Microsoft Network DDE 不允许建立任何新连接。

不过， **Wonderware NetDDE Helper** 服务也会监听和捕获 Microsoft Network DDE 的注销消息，以保持 Microsoft Network DDE 连接。因此，在您注销时， Microsoft Network DDE 允许建立新连接。

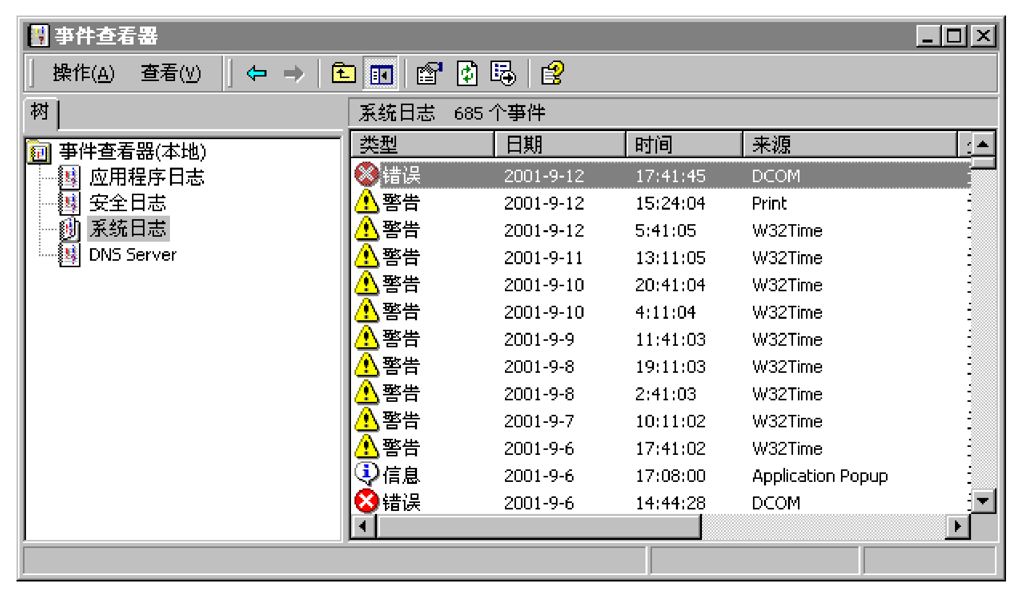
如需有关详细信息，请参阅 “在桌面和系统环境下运行 Windows NT 服务”。

排解 InTouch 服务疑难

以下是 “Wonderware 技术支持”在安装或运行 InTouch 服务期间遇到的一些常见问题及其解决方案：

错误 “一个或多个服务无法启动 ...”

如果您遇到此错误，请从 “开始”菜单选择**程序 / 管理工具 （公用） / 事件查看器**。此时会出现**事件查看器 - 系统** *日志*对话框，并显示一个列表，列出在启动任何 Windows NT 服务期间可能已出现的提示性消息、警告或错误。



您可以查看因 InTouch 服务无法启动而生成的任何警告或错误。如果**事件查看器**指出 Wonderware WindowViewer 服务无法启动，则大多数是因为该项服务所依赖的其它 InTouch 服务未在运行。

如需有关详细信息，请参阅 “各项 InTouch 服务之间的相关性”。

在配置 Wonderware WindowViewer 服务的运行选项之后，则必须将它当作服务来启动和停止。

如果从快捷方式图标或通过从 “开始”菜单选择**程序 /Wonderware FactorySuite/InTouch WindowViewer** 来启动 WindowViewer，则 WindowViewer 将暂停约 15 秒，然后才显示此消息：



单击**是**，可以让 Wonderware WindowViewer 服务作为您用户帐户下的桌面应用程序来启动，并显示其 GUI。

注意！ 如果选择**是**以便将 WindowViewer 当作桌面应用程序运行，则在注销系统时， WindowViewer 会关闭。由于它原来不是作为服务启动的，因而不会作为服务继续运行。这表示在您注销之后， InTouch 应用程序不会继续运行并收集历史数据、显示报警或处理脚本。

要让 InTouch 应用程序能够继续作为服务运行，则您需要重新登录，并将 WindowViewer 配置成作为 NT 服务来启动。

单击**否**以取消操作。此时 WindowViewer 应用程序将不会启动。

如需有关详细信息，请参阅第 2 章的 “将 WindowViewer 作为 NT 服务运行”部分。

InTouch 服务无法安装或启动

如果在安装 InTouch 后无法安装或启动 InTouch 服务，请执行以下步骤：

1. 打开 “Windows NT 用户管理器”窗口，并创建一个新的主用户帐户。

   备注 要将 InTouch 组件作为服务来启动，此用户帐户必须具有本地计算机的 “管理”权限。如果在域名列表中您看不到计算机的节点名，请手动输入节点名。

   如需有关详细信息，请参阅 “创建主用户帐户”。

2. 确认计算机的节点名不超过 14 个字符。如果节点名包含下划线字符 (_) 或短划线 (-)，请将节点名更改为不含下划线或短划线且等于或少于 14 个字符的名称。

3. 在安装期间，如果提示您输入域名，请输入计算机的节点名而不是域名。然后输入在步骤 1 中创建的用户名和口令。

4. 如果已经安装 InTouch，则您仍可通过从目录 C:\Program Files\Factorysuite\Common 运行 “Wonderware 服务用户”程序 wwuser.exe 来指定域名、用户名和口令。
5. 重新启动计算机。
6. 使用任何有效的用户帐户登录到网域。即使域已经关闭，它也不会影响到在本地计算机上运行的 InTouch 应用程序。

InTouch 服务的注册表项

所有四个 InTouch 服务均在 Windows NT 注册表中作为注册表项列出，具体如下：

InTouch 服务注册表项

Wonderware SuiteLink
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SLS

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\slssvc

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SuiteLink

Wonderware Logger
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWLOGSVC

Wonderware NetDDE Helper
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWNetDDE

Wonderware WindowViewer
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VIEW

词汇表

ActiveX

一种松散定义的技术集，可供软件组件彼此交互作用。

ActiveX 控件 / 容器

ActiveX 控件的前身是 OLE 控件或 OCX，它们是以标准方式执行特定功能的独立软件组件。它们为可重复使用的组件定义标准的接口。 ActiveX 控件不是独立的应用程序。相反，它们是放置到控件容器中的服务器。要使用 ActiveX 控件，必须将它们放置到 ActiveX 容器中。 InTouch 便是一个 ActiveX 容器。 VisualBasic 和 Internet 浏览器也是 ActiveX 容器。

Application Publisher

InTouch Application Publisher 可以创建自解压文件，该文件包含在另一 InTouch 节点上安装应用程序所需的全部相关文件及安装过程。Application Publisher 通过压缩 “应用程序”来最大程度减少从 Internet 进行下载所需的时间。

B

字节或波特的缩写。指存储时通常使用 “位”；指通讯时通常使用波特率。Kb = 1000 字节或波特（从技术上而言，1K = 1024 字节）。详见**波特**或**字节**。

Beta 测试

由所选的 “典型”用户对硬件和软件产品进行预测试，以便产品在投放到市场以前发现各种缺陷。

BIOS

Basic **I**nput/**O**utput **S**ystem （基本输入 / 输出系统）的首字母缩写形式。在一些操作系统中，指根据特定计算机进行自定义的程序部分。

CD-ROM

“压缩磁盘只读存储器”(Compact Disc, Read-Only Memory) 的缩写。可存放超过 650 MB 数据的压缩磁盘格式。

COM 端口

个人计算机上的串行通讯端口。

CONFIG.SYS

系统开启或重新启动时由 MS-DOS 处理的 ASCII 文本文件。它可供用户配置操作系统的某些方面，如分配的内部磁盘缓冲区数目、可同时打开的文件数量等。

CSV

“逗号分隔变量”是 “剪贴板”使用的一种格式，可用于在应用程序之间传送多列文本和数据值。 CSV 数据项类似于用逗号分隔每个变量的文本。尽管 Microsoft Excel 可能是创建 CSV 剪贴板数据的主要程序，但许多 DOS 应用程序也都支持这种格式。

DHCP

“动态主机配置协议” (Dynamic Host Configuration Protocol) 的缩写形式。这是用于自动分配 IP 地址，并给网络客户端配置 TCP/IP 的一种 TCP/IP 协议。

DRA

动态地址引用，可以通过使用点域，在运行时动态控制 I/O 标记名的访问名和项目。它还可供通过 QuickScript 函数来控制访问名属性。

DRC

动态分辨率转换，可以让每个 View 节点使用多种自定义选项 （包括自定义分辨率）来缩放应用程序。此种比例调整可以在 WindowViewer 编译应用程序时进行，并且不需要 WindowMaker。

ENTER 键

键盘上执行某个语句或命令的键。在有些键盘上相当于 RETURN 键。

FactorySuite

Wonderware 的软件包，包括 InTouch （及其所有的附加程序和实用程序）、InBatch、InTrack、InControl 及其 I/O Server、IndustrialSQL Server、FactorySuite Web Server、其它各种 I/O Server 程序、 Productivity Pack、NetDDE for Windows 以及 NetDDE Extensions for Windows NT。

HMI

人机界面。可供操作员控制制造过程的一种软件程序。也就是所谓的 MMI （Man-Machine Interface，人机界面）。例如 Wonderware InTouch。

I/O

Input/Output （输入 / 输出）的缩写。

ICA

独立计算架构。 Citrix Systems 开发的一种远程显示服务协议，可供瘦客户机访问服务器。

Internet

全部使用 TCP/IP 进行互联的庞大网络群体，由二十世纪六十年代晚期及七十年代早期的 ARPANET 发展而来。 Internet 将大约 70,000 个独立的网络连成一个全球网络。

IP 编号或地址

以句点分隔为四个部分的编号（例如 123.45.67.89），可唯一确定 Internet 上的某台机器。 Internet 上的每台机器都有一个唯一的 IP 编号；如果机器没有 IP 编号，则它不能出现在 Internet 上。大多数机器还有一个或多个域名，以方便人们记忆。

IP （Internet 协议）

用于网络通讯的公共协议。

Java

网页中使用的类似于 C 和 C++ 的高级编程语言，可提供动画及其它高级功能，以制作出风格独具的网页。

MB

Megabyte （兆字节）的缩写。表示一百万个字节。等于 1000KB。

Megabyte

实际上是 1,048,576 字节或 1024 千字节；或粗略地计算为一百万字节或一千千字节。

MS/DOS

MICROSOFT **D**ISK **O**PERATING **S**YSTEM （Microsoft 磁盘操作系统）的缩写，是 IBM “个人计算机”及兼容机使用的标准操作系统。由 Microsoft, Inc. 开发。

NAD

“网络应用程序开发”或 NAD 是一个特殊的结构，它同时结合了基于客户端和基于服务器结构的优点。 NAD 提供自动通知应用程序更新的功能，并具有将更新后的应用程序自动发布到 View 节点的功能。

NetBIOS 增强用户接口 (NetBEUI)

需要少量内存但不能进行转发的小型快速通讯协议。通过路由器链接的远程位置不能使用 NetBEUI 进行通讯。

NTFS 文件系统

Windows 2000 和 Windows NT 自身的文件系统。它支持长文件名、各种共享文件权限，以及事务处理日志；如果操作系统不慎中断，则可以使用事务处理日志来完成任何未完成的且与文件相关的任务。

QuickFunction

QuickFunction 指可以在其它脚本或表达式中进行编写和调用的脚本。每个 QuickFunction 最多可以使用 16 个参数。它们存储于创建它们的应用程序中。 QuickFunction 可定义为异步的，也就是说，执行时它在主 WindowViewer （运行时）进程的后台运行。

QuickScript

在 InTouch 中创建的脚本。 InTouch QuickScript 功能可供在满足指定条件时执行命令和逻辑运算。例如，键被按下、窗口被打开，值发生改变等。

RAM

随机访问存储器。计算机的主要存储空间。

SuperTag

InTouch 支持一种模板结构，可供定义 SuperTag 这种复合型标记名。SuperTag 模板最多可容纳 64 个成员标记名和 2 个嵌套级别。详见 **TemplateMaker**。

TemplateMaker

可供创建 SuperTag 模板的 InTouch 实用程序。详见 **SuperTag**。

TSAC

Microsoft 的 Terminal Services Advanced Client （终端服务高级客户端）是一个基于 Win32Æ 的 ActiveXÆ 控件，可用于在 MicrosoftÆ Internet Explorer 中运行 “终端服务”会话。通过使用 TSAC，现在您可以在 Internet 上运行全功能的 InTouch 应用程序，其性能和速度可以与在局域网上运行相媲美。

UNIX

一种计算机操作系统，设计由许多计算机用户同时使用 （多用户）且内置 TCP/IP。这是 Internet 上最常见的服务器操作系统。

Windows

Microsoft 开发的一个操作环境。

Windows 应用程序

专为 Microsoft Windows 操作环境设计的应用程序，它使用 Windows 的所有功能，如菜单、滚动条以及图标。

x 轴

坐标平面上的水平轴。

y 轴

坐标平面上的垂直轴。

按钮

在对话框中出现的大圆角矩形或小圆角的按钮。使用指针箭头单击可选择按钮的选项或命令。

帮助

介绍如何使用 Windows 应用程序的联机操作说明。“帮助”菜单会显示特定的 “帮助”主题。

保存

将文件或对文件所作的更改存储到磁盘上。

保留标记名

标记名的一种，它的值会被保留下来，并且在每次系统出现故障而重新启动之后被用作自己的初始值。

报警

只要出现严重偏离正常条件的情况，便会显示或激活的警告信号。

备份

创建指定的一个或多个文件的备份，将它们从一片软盘或一块硬盘传送到另一个可拆卸或固定的磁盘。

边界

包围当前活动窗口的线段。鼠标指针显示为双向箭头时，可以通过拖动边界来调整窗口大小。

标记名

给 “标记名字典”（InTouch 数据库）中定义的变量指定的名称。

标题栏

每个窗口顶部的横条，包含应用程序的名称与该应用程序所用文档或文件的名称。（**InTouch** 提供了一个可供隐藏 “标题栏”的选项）。标题栏也可用于在屏幕上移动窗口，方法是鼠标指向标题栏单击并拖曳。

表达式

描述数字、带运算符的数字、变量及上述各项值组合的一般术语：6、3+6、 n+10 均为表达式。

波特率

测量数据传输速度的单位。一波特等于每秒 1 位。因为单个字符要求使用 8 位来表示，因而将波特率除以 8 便得到每秒要传输的字符数 (cps)。例如，300 波特相当于 37.5 cps，1200 波特相当于 150 cps，2400 波特相当于 300cps 等。

不活动

未被选中的窗口或图标。详见**选择**。

裁剪

在计算机图形中，切除图像的某个部分。

菜单

菜单是可用的 Windows 和应用程序命令组的列表。菜单标题出现在窗口顶部的菜单栏中。通常您可以先显示菜单，然后再选择所需的命令。

菜单栏

列出应用程序菜单的名称的水平条。菜单栏显示在窗口标题栏的下方。虽然某些菜单 （和命令）在许多应用程序中都可以看到，但每个 Windows 应用程序都有它特定的菜单栏。

参数

可赋予逻辑或数字值的一种变量。最多可以给 InTouch QuickFunction 指定 16 个参数。详见 **QuickFunction**。

操作系统

控制执行计算机程序的软件，可提供计划任务、调试、输入 / 输出控制、存储分配等。通常缩写为 OS。

插入

用于将某个值放入 （插入）计算机存储器中特定位置的指令。

插入点

用户进行输入时将插入文本的位置。插入点通常显示为一个不断闪烁的垂直线 （光标），常会出现在工作区或对话框中。输入的文本出现在插入点左侧，输入文本时，光标会被推向右侧。

触摸屏

用户可以通过手指或其它物体触动指定区域来输入指令的显示屏。

处理

当前控制处理器的应用程序。收到消息时，应用程序便会取得处理器的控制权。它将一直控制着处理器直至消息处理完毕。

传输控制协议 /Internet 协议 (TCP/IP)

由网络用于同 Internet 上的其它网络进行通讯的协议。

串联

将两个或多个字符串联结成一个字符串，或将一行显示内容与下一行相连接。

串行端口

计算机中的一种输入 / 输出端口，通过它一次只能传输或接收一个比特的数据。大多数情况下，个人计算机是通过 RS232C 串行接口端口来传递串行数据的。

窗口

屏幕上供用户在其中查看和使用应用程序的一块矩形区域。屏幕上一次可以打开多个窗口，并且这些窗口可单独调整大小和移动。

磁盘操作系统

(DOS) 是一种操作系统，其中的操作系统程序存储在磁盘中。通常，它可以跟踪文件、存储和检索文件、分配存储空间，以及管理与磁盘存储有关的其它控制功能。

带宽

网络上通信通道的传输能力，以每秒兆位 (Mbps) 表示。例如，"以太网"的带宽是 10 Mbps。"高速以太网"的带宽是 100 Mbps。

当前文件

正在应用程序中运行的文件。

登录

进入计算机系统的行为，例如 "登录到网络并收阅电子邮件"。

登录名

用于访问计算机系统的帐户名。与口令不同，登录名是公开的。

电子表格

在单元格矩阵中处理数据和公式的程序。例如 Microsoft Excel。

调色板

计算机图形系统中的可用颜色集。

动态数据交换

DDE 是应用程序之间的数据通道，无须用户介入或监视即可完成。在 Windows 环境中， DDE 是通过一组消息类型、处理这些消息类型的建议程序 （协议），以及一些新定义的数据类型来实现的。按照这些协议，这些独立编写的应用程序可以相互传递数据，而无须用户介入。例如 InTouch 和 Excel。

动态主机配置协议

详见 **DHCP**。

独立式

一种自包含的计算机系统，与连接并依赖于远程计算机系统的计算机相反。独立式计算机可以自行操作，而不必依赖其它设备。

端口

外设可以通过它进行通讯的计算机部分。 CPU 与外设之间的连接。

队列

等待计算机去执行的一组项目。项目的排列方式确定了处理的优先级。例如，等待打印的文件。

对话框

执行某个命令之前 Windows 需要进一步的信息时所出现的窗口。例如，如果选择**文件**菜单上的**保存**命令，则会出现一个对话框，询问用以保存文件的名称。

对象

数据集。对象有几种格式：位图图像、文本、实时图表和历史趋势图等。

多任务

计算机同时操作两项或多项功能或任务的能力。

二 - 十进制 (BCD)

用一组 4 位二进制 **1** 和 **0** 来表示十进制数字的计算机编码系统。

二进制

一种以 2 为基数的数字系统；或一种仅包含两种可能的选项或条件的特征或属性。

二进制码

通过使用位 - 即 0 或 1 对数据进行编码的编码系统。

方法

方法与 ActiveX 控件有关。它们类似于可从 ActiveX 容器中进行调用的脚本函数调用。例如， **Browser.Navigate("URLPageName")**、**Engine.start()**。详见**属性**和**事件**。

访问

数据的获取。定位所需的数据。

服务

一种特殊类型的程序，它享有 “特权”并在系统内以很低的级别在运行。服务程序自动在后台运行，不要求用户登录。由于 Windows NT 操作系统是一种安全操作系统，普通的程序不允许直接访问硬件 （如硬盘）或其它系统对象 （如 “事件日志”）。服务程序可以替其它的普通程序访问硬件和系统对象。例如， Wonderware Logger、 WindowViewer 均可作为 NT 服务来运行。

服务器

指为网络上的其它计算机提供服务的计算机。例如，文件服务器便可以给客户端机器提供文件。

服务器群集

共担责任的一组互相连接的服务器，通常配置为在一个或多个服务器崩溃时可以继续执行处理。

赋值语句

一种编程语言语句，可以将值赋给变量，如在 *x = x + 1 或 y = 6* 中。

赋值运算符

赋值语句中使用的运算符，可以将运算符右边的值放入运算符左边的变量中。

复选框

对话框中出现的可打开或关闭的小矩形框。复选框通常与可设置的多个选项关联。要设置复选框选项，请移到它上面，然后单击鼠标按钮。出现 **X** 时，它便被选中。如果它是空白的，则表示未被选中。

负载平衡

配置服务器群集中的服务器，以便在各个服务器上平均分配负载需求。

高速以太网

详见以太网。

格式

准备磁盘以便存储信息的过程。格式化磁盘会擦除任何先前存储的数据。格式是用于表现对象的术语。在大多数 Windows 应用程序中，可用的格式包括 “文本”、“位图” 等。

工作区

显示应用程序本身及所有其它打开的窗口的应用程序窗口区域。

关闭

从屏幕上移除应用程序的窗口和图标，并释放该应用程序所占用的内存。要关闭应用程序，请选择*控件 / 关闭*命令。应用程序关闭之后，必须重新运行才能再次使用它。

滚动

上下或左右移动数据或文本，以查看未在屏幕上显示的文件部分。

滚动框

滚动条中的小白框。滚动框反映窗口内的信息相对于整个文件内容的位置。例如，如果滚动框在滚动条的中间，则窗口中的文本或数据即处在文件中间。您可以使用鼠标拖动滚动条中的滚动框来翻阅文件。详见滚动条。

滚动条

出现在窗口底部或右边的横条或竖条。在窗口包含的信息无法用一屏显示时，可以使用滚动条在其中移动。窗口右边的滚动条可以垂直滚动。窗口底部的滚动条可以水平滚动。

过程控制

指使用计算机来控制工业过程，例如石油精炼和钢铁生产。

过程控制计算机

过程控制系统中使用的计算机，通常在指令容量、字长和精确度方面会有一定的限制。设计用于在非空调环境下的连续作业。

毫秒

千分之一秒，简写为 ms 或 msec。

宏

一种符号化编程语言语句，可以转译成一系列的机器语言语句。

后台 / 背景

在多任务程序设计中执行低优先级程序的一种工作环境。此外，它也可以指显示屏中未被显示字符或图形 （前景）占据的部分。

缓冲区

一块存储区，用于临时存放从一个设备传输到另一设备的数据。它用于补偿各种硬件设备处理数据时的不同速率；例如，使用缓冲区存放等待打印的数据，以释放 CPU 给其它任务使用，因为它处理数据的速率更快。

辉亮显示

指出对象已经被选中，并且它会受到下一个动作或命令的影响。辉亮显示的对象会反白显示。选定的图标轮廓呈白色，并显示应用程序的名称。

恢复

通过双击可将图标恢复成最大尺寸的窗口的行为。要恢复窗口，请从 “控件” 菜单选择 “恢复” 命令，或单击窗口右上方的 “恢复” 框。详见**最大化**和**最小化**。

混合

使用不同颜色点的排列组合以形成一种新单色的方法。

活动应用程序

所创建的窗口当前拥有键盘焦点的应用程序。应用程序不必是活动应用程序即可接收并处理各种消息。只要应用程序获取或失去 “活动应用程序” 状态，便会有消息去通知它们。用户通常会确定去活动应用程序，但应用程序可以覆盖此决定。

寄存器

中央处理单元中使用的高速处理设备，用于在处理期间临时存储少量数据或中间性结果。

计算机图形

一个范畴较广的术语，概指显示屏或硬拷贝输出设备上显示的图形或图表之类外观的物体，与字母和数字相区别。

剪贴板

一块用于存放数据 （文本、位图、图形对象等）的存储区，这些数据可被复制或移动到另一应用程序或窗口。

键加速器

执行菜单命令的特殊键盘序列。例如 Ctrl + A。详见**快捷键**。

节点

目录树结构中的一个位置，它与自己下面的一个或多个项目相链接。在局域网上，指的是可以同网络上的其它设备进行通讯的一种设备。在群集环境中，指的是运行 Windows 2000 Advanced Server 且属于群集之一员的计算机。

镜像

显示或创建与原始方向完全相反的图象。例如，按 x 或 y 轴翻转图形。

局部变量

您可以在脚本中声明局部变量来存储临时结果，并使用临时脚本值来创建复杂的计算，而不会影响或减少授权的标记名计数，并可提高性能。在相同的脚本中，局部变量或标记名可以互换使用。

局域网 (LAN)

一组互相连接的电脑，通常彼此靠近 （如在同一幢建筑中或某幢建筑的同一层），以便可相互传递数据。

客户端

与服务器交换数据的机器。与服务器共享数据的软件程序。

客户端 / 服务器

客户端机器依赖于服务器的网络。

控件名

详见 **Windows 控件**

控制台

在服务器上运行的管理会话。

快捷键

快捷键由应用程序在创建键盘界面时使用。通常它们是作为使用菜单指示选项的替代方法而提供的。快捷键是一种击键，对于应用程序而言具有特殊的含义，并可用于生成命令消息。

扩展

在窗口中选择多个项目。要扩展选定内容，请按住 SHIFT 键直至选中所有的项目。

扩展名

文件名结尾的句点及三个字母。扩展名确定文件所包含的信息之类型。例如，扩展名 EXE、 BAT 表示文件包含一个应用程序。

离散值

仅有两种状态的一种变量：'1' (True, On) 或 '0' (False, Off)。

离线

指设备或装置不直接与计算机的中央处理单元进行通讯的状态。也指未连接到计算机的设备。

列表框

对话框中的一个框，列出命令的所有可用选项。例如，磁盘上文件名的列表。通常是从列表框选择某个项目，然后再选取 “确定”按钮。如果列表框中的选项太多，无法一次性全部显示，则会出现一个垂直滚动条。选择列表中第一个项目旁边的下箭头，则会显示列表框中其余的选项。

路径

文件的层次结构，通过它可以控制查找特定文件的路径。如果在当前或指定的驱动器或目录下未找到程序或批处理文件，则系统将指定一个或多个要按顺序搜索的磁盘驱动器和 （或）目录路径。系统按 “路径”中显示的顺序搜索驱动器和 （或）目录路径。

路径名

系统内目录或文件位置的描述。路径名包含盘符、冒号 (:)，后跟目录或子目录名，然后再跟文件名。各个名称与前面的名称之间用反斜杠 (\) 隔开。如果未加指定，则使用缺省驱动器和目录。

轮询

一些计算机 / 终端系统使用的通讯控制方式，通过它计算机可以依次询问连接到公共传输媒介的许多设备，以确定是否有要传递的信息。

命令

执行特定动作的词或短语，通常出现在菜单中。

命令按钮

一种圆角矩形，它上面有描述动作的标签，例如**确定**、**取消**或**关闭**。选中之后，命令按钮即可执行该动作。

命令键

用于执行不同功能的键盘键。

命令行

跟在任意 MS-DOS 命令后面的参数字符串，包含初始化应用程序的命令。命令行中的参数会传递给 MS-DOS 函数或在启动期间传递给程序。

模除

取除法余数的一个数学函数。数字 *x* 模除 *n* 的结果为 *x/n* 的整数余数。例如， 200 模除 47 等于 200/47 的余数，即 12。

模拟

指通过测量连续物理量来表示数量的一种表示法。

模式

操作的方法或条件。

目录

一种将文件分组以便存取的结构。目录就像是显示文件位置的地址。目录可以包含文件或文件子目录。

平铺窗口

平铺窗口是大小、形状及其在屏幕上的位置均由 Windows 确定的窗口。平铺窗口是唯一不与其它窗口相互重叠的一类窗口，它可以放入图标区并且可以带有菜单。每个应用程序通常只创建一个平铺窗口。应用程序创建的其它窗口通常为层叠或弹出式窗口。

奇偶校验

在字节、字符或字中额外添加的一位，以根据系统逻辑，确保总有奇数位或偶数位。如果因硬件故障而导致传输过程中丢失某个位，则可通过奇偶校验检测到丢位情况。只要字节、字符或字的内容不变，就可以保持相同的位组合。

企业

用于概述所有业务经营单位，包括所有远程办公室和分支机构的一个术语。

启动

通过从存储设备将指令读取到计算机的内存，以启动或重新启动计算机系统。它涉及到将操作系统的一部分加载到计算机的主内存中。如果计算机已经打开，则称它是“热启动”；否则，会称它是“冷启动”。

千兆以太网

详见**以太网**。

切换

指任何具有两个稳定状态的设备。与“触发器”是同义语。

驱动器

A-Z 范围内的一个字母，后跟冒号 (:)，以指示逻辑磁盘驱动器。

全限定域名 (FQDN)

包括回溯到根节点的所有网络域名称的一种域名，如此一来便可以明确地指出域名空间目录树中的某个位置。 FQDM 的一个例子是 Sales.Europe.Wonderware.com。

缺省值

系统自动选择的选项、命令或设备。例如，在打开对话框时，对话框中的某个命令按钮已被选取。这表明它是缺省值，如果按一下 **<Enter>** 键，则它会被自动选取。通过选择另一个适当的选项、命令或设备，可以改写缺省值。

任务

任务是正在执行的应用程序。任务是 “进程”的同义语。

设备驱动程序

控制计算机与打印机、显示器或鼠标等设备进行交互作用的程序。通过使用设备驱动程序，计算机才能使用这些设备。

时间间隔

时间单位。

事件

事件与 ActiveX 控件关联并通过 ActiveX 容器发生。您可以设计某个动作，并通过创建 “ActiveX 事件脚本”将它与事件关联，以便在运行时 (WindowViewer) 执行 ActiveX 控件事件。例如， **Control.click (shift)**。**FileViewer.DoubleClick (name)**。详见**属性**和**方法**。

视图区域

Windows 应用程序中的视图区域 （也被称作 “工作区”）会显示文件的一个页面。详见**工作区**。

瘦客户机

不配硬盘且可访问服务器的终端。

属性

属性与 ActiveX 控件关联，并可以同 InTouch 标记名关联。可以给特定的 ActiveX 控件配置的属性是由 ActiveX 控件的设计人员确定的。有些属性是单向的，这表示或者是该属性去设置标记名的值，或者是标记名的值去设置该属性。而其它属性均是双向的，这表示值机可以从标记名也可以从属性去设置。

数据包

通过网络发送的信息的基本单位。每个数据包均包含目标地址、发送者地址、错误控制信息以及数据。数据包的大小和格式取决于所使用的协议。

数据库

逻辑上相关的记录或文件的集合。数据库将许多记录合并到一起，形成一个公共的数据记录集合，并将它当作一个核心文件来使用。

算法

一系列的指令，它们机械化地执行，以完成特定的过程。

缩放

改变图像大小的过程。

同步传输

以固定速率传输各个比特的数据传输。为了保证同步，传输器和接收器均使用相同的时钟信号。详见**异步**。

图形对象

一种视觉化对象，如用来表示可视化界面的滚动条、位图或图标。图形对象可以使用应用程序来创建，也可以由 Windows 来创建供该应用程序使用。

位

二进制数；数字的二进制表示法中的一位 （1 或 0）。这是计算机及其相关设备所能识别的最小信息单元。多个位可以组成一个字节或计算字。

位图

一部分显示设备表面的内存映像。在 Windows 中，位图实际上是一种数据结构，它包含指向此内存映像的指针，以及关于显示设备的信息。位图所需的内存量与设备有关，它取决于所涉及的设备的颜色能力和象素分辨率。

文本框

输入执行命令所需信息的一个框。文本框通常出现在对话框中。

文档

必须连续打印的一种打印机输出单位，这就是说，其它输出不能穿插到文档中。另一方面，也可以将文档称作报告。应用程序必须制定每个文档的开头和结尾。

文件

将信息保存放并存储到硬盘或软盘以备日后使用的一种机制。文件也指用户创建的任何文档或数据库，如字处理文档、电子表格等。每个文件显示在它自己的窗口中，在大多数情况下，文件名会显示在窗口顶端的标题栏中。

文件名

文件名由不超过 8 个字符的基本名称和三个字符的扩展名组成。例如 INTOUCH.EXE。

显示

数据在屏幕上的物理表示法。

象素

图形单元格。“图形元素”的简称。可视化的显示屏按行和列分成极小的点、方形或单元格，其中每个都是一个象素。显示屏网格上可以存储或显示的最小单位。计算图形通常由矩形象素方阵组成。图形的分辨率通常用显示屏中的象素个数来表示。例如，560 x 720 象素的图形要比 275 x 400 象素的图形清晰得多。

消息框

一种特殊的对话框，通过它应用程序可以显示错误消息或其他重要的信息。发生错误或应用程序需要信息来完成某个特定的动作或命令时，消息框可以提醒用户。

协议

控制计算机系统或应用程序之间进行信息交换的一组规则或管理。

星号

星号 (*) 在许多编程语言中用于表示乘法运算符。

选项按钮

对话框中显示的一个圆形小按钮。通过选择选项按钮可以设置选项，但在一组相关的选项按钮中，只能选取其中一个。选中某个选项按钮时，它会显示一个黑点；如果未被选中，则显示为空白。

选项按钮组

对话框中一组相关的选项。一次只能从按钮组中选取一个选项按钮。

页面

页面指选定并存储在文件中的信息块。例如， Microsoft Word 的一段文字可以是一个页面， Microsoft Excel 的图表可以是另一个页面。在同一个文件中，页面可以有各种格式。在放入文件时，页面会被编号。

已发布的应用程序

在服务器群集中的各个服务器之间平等共享的应用程序。

以太网

施乐公司在 1976 年开发的一种局域网协议。以太网支持 10 Mbps 的数据传输速率，它采用总线拓朴逻辑，并使用粗细不等的同轴电缆、光纤或双铰线进行传输。新版 “以特网” 被称作 “高速以太网”，支持高达 100 Mbps 的数据传输速率，而更新的版本 “千兆以太网” 则更支持高达 1000 Mbps 的数据传输速率。

异步

属于一种数据通讯模式，可以在传输过程中于字符之间提供可变的时间间隔。详见**异步传输**。

应用程序

用于执行特定工作的程序或程序组，例如 InTouch。

优先权

确定应该先执行表达式中的哪个运算符的规则。

语法

规定语言结构及其表达式的规则。

语句

计算机语言中的指令表达式。

域

在 Windows 2000 中，指共享安全策略和用户帐户数据库的一组计算机。Windows 2000 域与 Internet 域不同。

域名

在 Active Directory 中，指给共享公共目录的联网计算机集合指定的名称。在 Internet 上，这指的是可以唯一确定特定主机的文本名称。一台机器可以使用多个域名，但给定的域名只能指向一台机器。域名可以通过 DNS 域名服务器解析为 IP 地址。

域名空间

按照命名惯例定义的名称或名称组；可解析指定名称的任意有限区域。Internet 使用层次结构化的域名空间将各个名称划分成不同类别 （称为顶级域名），如 .com、 .edu 以及 .gov。

域名系统 (DNS)

TCP/IP 网络 （包括 Internet）上将域名解析为 IP 地址的一种服务。这样，在查询远程系统时，用户可以采用便于记忆的名称如 FinanceServer 或 Ourbusiness.com，而不必使用 123.45.67.89 之类的 IP 地址。

远程标记名

存在于远程标记源但却在本地 InTouch 应用程序中引用的标记名。通过远程标记名引用，无需使用本地 “标记名字典”中的任何标记名便可设计客户端应用程序。

远程桌面协议 (RDP)

随 Windows Terminal Service 一起安装的缺省连接协议。 RDP 可供您给每个服务器配置各种不同的设置。客户端通常控制这些设置中的大部分。

运算符

在描述过程时，指要在运算数上执行的动作。

运算数

用以执行运算的数量或数据项。

运行

启动应用程序。“运行”命令允许您给应用程序指定参数。通过双击名称或图标也可以运行应用程序。

运行时

控制单元获取数据并在算术逻辑单元中执行实际处理的时间。也指程序执行的时间。

运行中

“运行中”的应用程序在系统内作为任务在运行的应用程序，它可以接收消息并且通常为用户所知。从初始化到终止，应用程序始终处在运行状态，但不总是在*进行处理*。详见**处理**。

粘贴

将某些内容从 “剪贴板”插入文档或文件。某些应用程序 （包括 InTouch）可能会包含一个执行此任务的 “剪贴”命令。如果在窗口中运行其它某个标准的应用程序，则 Windows 会将 “粘贴”命令添加到 “控件”菜单中。

整数

由正整数、负整数和零组成的集合中的任意成员。例如：-59、-3、 0。

终端

可供发送命令给另一部计算机的设备。最简单的终端通常是键盘、显示屏和一些简单的电路。个人计算机中通常使用终端软件，这种软件模拟物理终端，可供您给另一部计算机输入命令。

终端服务高级客户端

详见 **TSAC**

终端服务器

配备处理终端数据的多用户操作系统的服务器。

子目录

子目录位于目录。它们是将文件分组以便存取的一种结构。子目录类似于显示文件位置的地址。

字符串

作为单个数据项连接的一系列字符或位。

字节

一组相邻的二进制数，由计算机用作一个单元的。最常见大小的字节包含 8 个二进制位。

总线

传输数据的通道或路径。

最大化

使某个窗口或图标布满整个屏幕。要最大化窗口，请选择*控件 / 最大化*命令，或单击窗口右上角的 “最大化” 框。详见**最小化**和**恢复**。

最小化

将窗口转换成图标。要最小化窗口，请选择*控件 / 最小化*命令，或单击窗口右上角的 “最小化” 框。详见**最大化**和**恢复**。

索引

符号

B

C

D

E

F

G

H

J

K

L

M

N

O

P

Q

R

S

W

X

Y